P. 1877/71

10 1971

informatyka

SPIS TREŚCI

Str.

Przed VI Zjazdem Partii 1

Adam Huculak — „Automatyzacja zarządzania w przemyśle i jej przykłady w Związku Radzieckim" 2

Zbigniew Ładoś i **Sławomir Trautman** — „Z praktyki programowania systemów API" 8

Jowita Koncewicz — „SNOBOL — język do przetwarzania napisów" . . 11

J.Nievergelt i **J. C. Farrar** (tłum. **Janina Rowińska**) — „Co maszyny mogą, a czego nie mogą robić?" 17

WIADOMOŚCI PKAPI

„Plany krajowego przemysłu komputerowego na Plenum PKAPI" . . . 20

DYSKUSJE 23

Z KRAJOWEGO BIURA INFORMATYKI . . . . . III okł.

PRZEGLĄD WYDAWNICTW . . . . . 7, II, III i IV skrz.

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT
Warszawa
Czackiego 3/5

KOLEGIUM REDAKCYJNE
Redaktor naczelny prof. dr Leon ŁUKASZEWICZ

Doc. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), Władysław KLEPACZ, dr Antoni MAZURKIEWICZ, inż. Dorota PRAWDZIC (zast. redaktora naczelnego), dr inż. Andrzej TARGOWSKI

Sekretarz Redakcji mgr Wanda KAČER **Redaktor techniczny** Leszek ZAGAŃSKI

RADA PROGRAMOWA

Mgr inż. Jan Bursche, mgr inż. Henryk Chyrek, (wiceprzewodniczący) mgr inż. Ryszard Dąbrówka, mgr inż. Bolesław Gliksman, mgr inż. Józef Knysz, prof. dr Leon Łukaszewicz, mgr inż. Jan Matejak, prof. dr Tadeusz Peche (przewodniczący), mgr inż. Jerzy Trybulski (wiceprzewodniczący), dr Tadeusz Walczak, mgr Kazimierz Wasilewski, mgr Waldemar Wiśniewski (sekretarz), mgr Stefan Wojciechowski, dr inż. Henryk Woźniacki, mgr inż. Jan Zdzisław Żydowo

Redakcja: Warszawa, ul. Emilii Plater 20 m. 15, tel. 21-13-91. Zastępca redaktora naczelnego tel. 28-37-29

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka". Z. 2. Zam. 536. Papier powlekany V kl. 80 g. Obj. 3 ark. druk. Nakład 3200. U-90.

Cena egzemplarza zł 8.— **INDEKS 36707** **Prenumerata roczna zł 96.—**

Informatyka

dawniej Maszyny Matematyczne

zastosowania w gospodarce, technice i nauce

Nr 10

MIESIĘCZNIK

1971

ROK VII

Październik

ORGAN KRAJOWEGO BIURA INFORMATYKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ

Przed VI Zjazdem Partii

„O dalszy socjalistyczny rozwój Polskiej Rzeczypospolitej Ludowej"

Z wytycznych KC PZPR na VI Zjazd Partii przedłożonych pod ogólnopartyjną i ogólnonarodową dyskusję

„Naczelnym zadaniem VI Zjazdu będzie wytyczenie kierunków i zadań rozwoju społeczno-gospodarczego kraju, wynikających z aktualnych i perspektywicznych potrzeb i możliwości, zapewniających przyspieszenie postępu kraju i podporządkowanych realizacji podstawowego celu naszej socjalistycznej gospodarki, którym jest zapewnienie wszystkim ludziom pracy dostatniego życia, wyzwalanie twórczych sił naszego socjalistycznego narodu i rozwój jego kultury.

Sformułowaniu takiego programu sprzyja inicjatywa i energia twórcza oraz obywatelska troska o sprawy Polski, przejawione z wielką siłą przez klasę robotniczą, chłopów i inteligencję po VII i VIII Plenum KC, które przywróciły zgodność poczynań kierownictwa z odczuciami i pragnieniami społeczeństwa..."

Bardzo ważną rolę odegrało m.in. to, że: ...„Wzrosła rola i zmienił się charakter inteligencji w naszym kraju. Jej rozwój i nowy skład społeczny, nowoczesna struktura zawodowa oraz podnoszący się poziom kwalifikacji są rezultatem socjalistycznej rewolucji, industrializacji i dynamicznego rozwoju społeczno-ekonomicznego kraju. Inteligencja związana jest silnie z socjalizmem, służy socjalistycznemu rozwojowi i w nim widzi realizację swych aspiracji. Ludzie twórczej pracy umysłowej odgrywają coraz większą rolę w naszym życiu. Będzie ona wzrastać wraz z postępem rewolucji naukowo-technicznej. Istnieje obiektywna potrzeba dalszego liczebnego rozwoju inteligencji pracującej, podnoszenia jej społecznego znaczenia i wzrostu jej wkładu w postęp społeczno-gospodarczy i naukowo-techniczny kraju. Przez właściwe wypełnianie zadań związanych z nowymi potrzebami spełniać ona będzie swą społeczną powinność wobec narodu..."

GŁÓWNE KIERUNKI ROZWOJU PRZEMYSŁU

„Decydujące znaczenie dla przyspieszenia postępu kraju, rozwiązania jego problemów socjalnych, wzrostu stopy życiowej ludności posiada przemysł..." ...„Szybszy postęp techniczny i technologiczny jest dla nas koniecznością i musi to obowiązywać cały przemysł." „Przemysł musi zacieśnić współpracę z własnym i ogólnokrajowym zapleczem naukowo-badawczym, a w szczególności szybko wdrażać osiągnięcia nauki do produkcji... Należy maksymalnie popierać innowacje techniczno-ekonomiczne, duże i małe, dając im zielone światło w działalności przemysłowej. Należy też możliwie najpełniej wykorzystywać współpracę z krajami socjalistycznymi i kooperację z firmami zagranicznymi w celu przyspieszenia postępu technicznego i przejmowania obcych innowacji techniczno-ekonomicznych."

...„Szybszy niż przeciętny wzrost produkcji nastąpi w takich dziedzinach, jak przemysł motoryzacyjny, maszyn rolniczych, elektroniczny i maszyn matematycznych..."

...„Niezbędne jest kompleksowe opracowanie programu rozwoju elektroniki, uwzględniającego poglądy ekspertów w tej dziedzinie i doświadczenia światowe. Partia zwraca się do elektroników, uczonych, specjalistów i praktyków o szeroki współudział w rozwiązywaniu tego doniosłego problemu. W oparciu o odpowiednie badania i ekspertyzy, Rząd powinien opracować program rozwoju elektroniki w Polsce".

W latach 1972—1980 m.in. należy:

„...przyspieszyć modernizację i rozbudowę kablowych i radioliniowych sieci teletransmisyjnych z uwzględnieniem potrzeb rozwojowych telefonii, telegrafii i telewizji oraz transmisji danych..."

USPRAWNIENIE I UNOWOCZEŚNIENIE SYSTEMU GOSPODAROWANIA I KIEROWANIA

...„Realizacja podstawowych celów i zadań planu pięcioletniego na lata 1971—1975 oraz po roku 1975 jest w dużym stopniu uzależniona od konsekwentnego wprowadzania w życie reformy systemu gospodarowania... Reforma systemu gospodarki i państwa powinna zagwarantować (m.in.) —

...wdrażanie do praktyki osiągnięć postępu naukowo-technicznego i powszechne wprowadzenie wszelkich korzystnych innowacji techniczno-ekonomicznych oraz najnowszych osiągnięć w organizacji zarządzania, a w szczególności w zakresie informatyki..."

„Podstawowe znaczenie ma ujmowanie w planach 5-letnich kompleksowych programów, wymagających zharmonizowanego działania różnych resortów i organizacji. Programy te powinny być opracowywane np. dla chemizacji gospodarki, motoryzacji, elektroniki... środków technicznych dla potrzeb zarządzania itd...."

„Programy te należy koordynować i harmonizować na szczeblu centralnym i resortowym dla zapewnienia dynamicznego i zrównoważonego rozwoju społeczno-gospodarczego całego kraju. Powinno się to dokonywać przede wszystkim w drodze unowocześnienia metod bilansowania planu, wprowadzania metod optymalizacyjnych, analizy przepływów międzygałęziowych oraz wprowadzenia elektronicznej techniki obliczeniowej w celu szybszego opracowywania poszczególnych wariantów planu..."

„Unowocześnienie organizacji i zarządzania wymaga rozwinięcia i wdrożenia metod informatyczno-decyzyjnych w oparciu o wykorzystanie nowoczesnych technik przetwarzania danych".

„Propozycje w powyższych dziedzinach przedstawi komisja partyjno-rządowa dla unowocześniania systemu funkcjonowania gospodarki i państwa. Ogromne znaczenie dla opracowania końcowych propozycji w tym zakresie będzie miała dyskusja przedzjazdowa. Partia zwraca się do wszystkich pracowników gospodarki narodowej... o aktywny udział w tej dyskusji..."

ADAM HUCULAK
Instytut Maszyn Matematycznych
Warszawa

62+66/69(47):658.011.56

Automatyzacja zarządzania w przemyśle i jej przykłady w Związku Radzieckim

Omówiono cele, znaczenie i metody automatyzacji systemów zarządzania w przemyśle. Przedstawiono podmiotowe i przedmiotowe aspekty funkcjonalnej i proceduralnej strony zautomatyzowanych systemów zarządzania. Podano przykłady systemów wdrażanych w przemyśle radzieckim.

Działalność gospodarczą można utożsamiać z ciągłym procesem wzajemnie współzależnych decyzji. Decyzje te mogą dotyczyć różnych dziedzin, poczynając od perspektywicznego rozwoju gospodarki jako całości, a na bieżącej działalności poszczególnych jednostek kończąc.

Powodzenie działalności gospodarczej zależy od trafności decyzji. Trafność decyzji uwarunkowana jest zarówno czynnikami subiektywnymi, jak obiektywnymi. Czynnikiem subiektywnym jest kadra kierownicza. Uwarunkowaniem obiektywnym jest zespół zjawisk określanych mianem bariery organizacji zarządzania. Podstawowymi komponentami tego zespołu zjawisk są cztery rodzaje ograniczeń: strukturalne, kompetencyjne, motywacyjne i poznawcze.

Ograniczenia strukturalne wynikają z podziału zadań. Podziały strukturalne wyznaczają ograniczenia kompetencyjne. Przeszkodę w podjęciu prawidłowej decyzji mogą stanowić czynniki motywacyjne, jeżeli cel działania organizacji nie jest zgodny z interesem osobistym decydującego.

Szczególnie interesującymi czynnikami, które mogą wywołać istotne zakłócenia, są ograniczenia poznawcze. Pojawiają się one wtedy, kiedy osoby zajmujące stanowiska kierownicze nie dysponują informacjami niezbędnymi do podjęcia właściwej decyzji bądź też uzyskiwanych informacji nie są w stanie należycie spożytkować. Informacja może się bowiem okazać niepełna, spóźniona lub nieadekwatna do potrzeb danego stanowiska decyzyjnego.

Przeprowadzone ostatnio w ZSRR badania, którymi kierował członek Akademii Nauk ZSRR W. Trapeznikow wykazują, że w jednym resorcie 90% krążących informacji było zbędnych (ich przeznaczenia nikt nie umiał określić), brakowało natomiast informacji koniecznych na wielu stanowiskach decyzyjnych [1]. Nasilenie specjalizacji i kooperacji współczesnej produkcji przemysłowej doprowadza do złożonych wielostronnych organizacyjnych związków i współzależności między przedsiębiorstwami oraz w ich ramach. Związki te rzutują na wielostopniowość zarządzania. Na tym tle pojawia się konieczność tworzenia systemów zarządzania.

Jeśli uwzględnimy, że zarządzanie produkcją związane jest z otrzymywaniem, obiegiem i przetwarzaniem ogromnej ilości informacji w wyjątkowo krótkim czasie, to staje się rzeczą jasną bardzo ważne znaczenie, jakiego nabiera automatyzacja zarządzania z zastosowaniem komputera. Maszyny bowiem odkrywają nowe możliwości tworzenia elastyczniejszych i efektywniejszych zautomatyzowanych systemów zarządzania przedsiębiorstwami oraz ich zespołami.

Zautomatyzowane systemy zarządzania przemysłem usuwają ograniczenia poznawcze i rozszerzają świadomość osób podejmujących decyzje.

Związek Radziecki wszedł w okres przyspieszonego wprowadzania nowoczesnych metod organizacji i zarządzania przemysłem z pomocą komputerów.

1. ZAUTOMATYZOWANE SYSTEMY ZARZĄDZANIA W PRZEMYŚLE, ICH CEL I ZNACZENIE

Przez zautomatyzowany system zarządzania (ASZ) w przemyśle rozumiemy zorganizowany zbiór środków materialnych (komputer, oprogramowanie, urządzenia pomocnicze itp.) i ludzkich współdziałających w dążeniu do zautomatyzowanego, szybkiego i dokładnego, prawie bezbłędnego i niezawodnego przekształcania danych gospodarczych na informacje bardziej złożone i nieodzowne do efektywnego zarządzania przemysłem.

Celem ASZ jest usprawnienie zarządzania, jego ułatwienie i przyspieszenie różnorodnych i wieloszczeblowych procesów przetwarzania informacji oraz podejmowania decyzji z pomocą komputerów.

Znaczenie systemów i komputerów polega na zdolności rozróżniania i wybierania wariantów, możliwości obliczania z dowolną precyzją, jednoczesnego pamiętania dużych zbiorów informacji, na dysponowaniu bardzo dużą szybkością operacji oraz wysokim stopniem niezawodności i bezbłędności.

Ograniczoność systemów polega głównie na tym, że można rozwiązywać tylko takie problemy, które dają się zalgorytmizować i wyrazić w postaci rozkazów; można realizować tylko takie operacje, które uprzednio przewidziano w programie.

Systemy są bardzo racjonalną symbiozą zdolności ludzkich i możliwości komputera. ASZ przedstawia złożony kompleks ekonomiczno-matematycznych metod zarządzania i technicznych środków ich realizacji.

2. DWIE STRONY SYSTEMÓW — FUNKCJONALNA I PROCEDURALNA

Zautomatyzowany system zarządzania opracowuje się dla określonego podmiotu gospodarczego. Od systemu żądamy, aby spełniał on określony cel czyli określoną przez podmiot funkcję. Podmiotem gospodarczym może być przedsiębiorstwo, branża lub gałąź przemysłu.

Ten aspekt systemu nazywamy jego s t r o n ą f u n k c j o n a l n ą. Wykorzystujemy przy tym zalety komputerów. Drugim aspektem każdego systemu jest jego s t r o n a p r o c e d u r a l n a. Strona proceduralna stanowi spojrzenie od warsztatu i zmierza do zapewnienia poprawnej formy działania systemu. Sprowadza się ona do pokonania ograniczeń poprzednio wymienionych. Strona proceduralna jest środkiem prowadzącym do celu, którym jest uzyskanie informacji od systemu, czyli spełnienie przez niego określonej funkcji.

Użytkownika (przedsiębiorstwo) interesuje strona funkcjonalna, tj. cel systemu. Umiejętności twórców systemu przejawiają się w doskonałości proceduralnej.

Rozpatrując obie strony systemów, postaramy się naświetlić je pod dwoma kątami widzenia. Raz będziemy patrzeć na zagadnienie od strony podmiotowej, drugi raz — od strony przedmiotowej.

Od strony funkcjonalnej systemów, z punktu widzenia podmiotowego interesuje nas, czyje potrzeby zaspokaja system, natomiast z punktu widzenia przedmiotowego chcemy wiedzieć, co jest treścią zautomatyzowanego zarządzania, czyli jakie sfery działania obsługuje system.

Od strony proceduralnej systemów, z punktu widzenia podmiotowego interesuje nas, kto organizuje prace nad systemami i kto je opracowuje, natomiast z punktu widzenia przedmiotowego chcemy wiedzieć, jak przedstawia się zapewnienie środków technicz-

nych dla systemów — hardware — i jak zabezpiecza się oprogramowanie — software.

3. PODMIOTOWY ASPEKT FUNKCJONALNEJ STRONY SYSTEMÓW

Celem, dla którego tworzy się zautomatyzowane systemy zarządzania przemysłem w ZSRR jest udoskonalenie form zarządzania. Dąży się do wykorzystania możliwości nowej techniki. Wykorzystanie to ma miejsce w określonych podmiotach gospodarczych i służy do usprawnienia różnych dziedzin zarządzania. Podobnie, jak decyzje podejmowane są na różnych szczeblach hierachicznej struktury zarządzania, systemy zarządzania też tworzą swoistą „drabinę" kompetencji w granicach „pola decyzyjnego", wynikającą z podziałów strukturalnych.

Podstawowa linia podziału przebiega prostopadle do linii hierarchicznej, tj. pionowej.

W tym aspekcie w ZSRR dają się wyróżnić systemy:

- przedsiębiorstw
- zjednoczeń (branż)
- gałęzi (ministerstw)
- ogólnopaństwowe (GOSPLAN, GOSBANK, GOSSNAB itp.) [2]
- przewidywany supersystem (największy w świecie system „człowiek—maszyna" [3].

Drugą linię podziału można przeprowadzić poziomo w ramach przedsiębiorstwa. Interesującą klasyfikacją tego typu systemów dla przedsiębiorstwa przemysłu chemicznego podają autorzy Dygnikow i inni [4]:

1) zarządzanie technologicznymi agregatami, reaktorami, z których każdy jest wyposażony w dostosowane urządzenie sterujące

2) zarządzanie wydziałami produkcyjnymi, urządzenia pracują w warunkach doradcy dyspozytora; przykładem tego typu urządzenia jest KISŁOROD 1

3) zarządzanie całym procesem chemicznym, jak np. produkcją amoniaku lub inną wielofazową; przykładem jest system KASKAD

4) zarządzanie całością przedsiębiorstwa chemicznego.

Użytkownicy systemów — gałęzie i branże

Liczba ukazujących się publikacji dowodzi, że w ZSRR największe zainteresowanie komputerami wykazują przemysły: maszynowy, hutniczy i chemiczny. Kolejne miejsce zajmują przemysł spożywczy i górnictwo.

Automatyzacja zarządzania w gałęziach przemysłu jest dość ściśle związana z gospodarczym znaczeniem i technologią przemysłu. Uwzględniając decydujące różnice technologiczne, całość zagadnień zarządzania przemysłem w ZSRR można podzielić na cztery podstawowe grupy:

1. Przemysł ciężki stosujący procesy ciągłe

2. Przemysł ciężki stosujący procesy periodyczne

3. Przemysł ciężki o produkcji małoseryjnej i jednostkowej

4. Przemysł lekki.

Przedmiotowy zakres zastosowań automatyzacji w zarządzaniu jest uwarunkowany branżowo.

Specyfiką ZSRR wynikającą z wykorzystania warunków ustrojowych jest dążenie do udoskonalania zarządzania gałęzią lub branżą przemysłu. Pod kątem widzenia branży rozwiązuje się w pierwszej kolejności następujące problemy:

1. Unifikację dokumentacji, symboliki, nomenklatur itp.

2. Ujednolicanie gospodarki normatywnej

3. Ujednolicanie branżowych form organizacyjnych rozrachunku gospodarczego

4. Operatywną kontrolę charakterystycznych wskaźników

5. Porównawczą analizę techniczno-ekonomiczną przedsiębiorstw

6. Ewidencję, analizę i manewrowanie zasobami mocy produkcyjnych

7. Optymalizację planowania produkcji.

Treścią gałęziowych systemów zarządzania bywa najczęściej zintegrowane zarządzanie gospodarką materiałową, surowcową lub zbytem.

4. PRZEDMIOTOWY ASPEKT FUNKCJONALNEJ STRONY SYSTEMÓW

Z punktu widzenia przedmiotowego, tj. z punktu widzenia obszarów zarządzania, możemy wyodrębnić trzy typowe sfery: zarządzanie strategiczne, taktyczne i operatywne.

Sfery zarządzania różnią się między sobą dystansem czasowym, z którego podejmujemy decyzje, horyzontem czasowym i zakresem rzeczowym. W różnych sferach występuje różne nasilenie różnych funkcji zarządzania.

W sferze zarządzania strategicznego ustala się cele działania całego systemu, akcentuje się rachunek ekonomiczny i decyzje (rozbudowa gałęzi, rekonstrukcja branż itp.).

W sferze zarządzania taktycznego inicjuje się działania zapewniające realizację wspomnianych celów, określa się podmioty zapewniające wykonanie zadań i nieodzowne środki. Ustala się przedsięwzięcia organizacyjne optymalizujące efekt działania (określanie programu produkcyjnego itp.).

W sferze zarządzania operatywnego bieżąco reguluje się przebieg procesów gospodarczych.

Automatyzacja planowania i zarządzania strategicznego

Przez automatyzację strategicznego planowania i zarządzania będziemy rozumieli automatyzację funkcji zarządzania na wyższych szczeblach. Dotyczy to zagadnień perspektywicznych i podstawowych względnie długofalowych prognoz optymalizacji działalności branży. Na szczeblu przedsiębiorstwa będą to plany wieloletnie.

Poniżej podajemy dla ilustracji kilka przykładów zastosowań komputerów do rozwiązywania tego typu problemów.

Klujew [5] pisze o długoterminowym prognozowaniu zapotrzebowania energii (na okres 10÷20 lat) jako temacie do rozwiązania na komputerze. Tego rodzaju perspektywiczne planowanie ASZ jest niezbędne do określenia optymalnej strategii rozwoju energetycznych mocy przedsiębiorstw, ponadto potrzebne jest do optymalizacji bilansów paliwowo-energetycznych przedsiębiorstw, terytorialnych kompleksów produkcyjnych i regionów ekonomicznych.

Ten sam autor uzasadnia celowość krótkookresowego prognozowania (3÷5 lat) zapotrzebowania na energię elektryczną.

Innym zagadnieniem z pogranicza strategii i taktyki jest optymalizacja wielkości i rozmieszczenie zabezpieczającego zapasu urządzeń oraz części zapasowych w systemie energetycznym [6]. Przy scaleniu zapasu dla kilku przedsiębiorstw, ogólna ilość urządzeń znajdujących się w zapasie może ulec zmniejszeniu.

Strategiczne zarządzanie jest bardzo skomplikowane. Stosunkowo łatwiejsze i dlatego bliższe zrealizowania jest ono w przemysłach o masowej i jednorodnej produkcji. Trudności, w miarę zwiększania się liczby asortymentów i zmniejszania długości serii rosną. Komplikacje i trudności automatyzacji strategicznego planowania i zarządzania wynikają z konieczności wyprzedzającego poznania zupełnie nowych obszarów praw przyrody i życia społecznego.

Automatyzacja zarządzania taktycznego

Sfera zarządzania taktycznego obejmuje głównie planowanie bieżące, produkcję *sensu largo* (zaopatrzenie, produkcja *sensu stricte*, zbyt) i ewidencję. Zarządzanie taktyczne wiąże się głównie z czynnościami organizatorskimi, zmierzającymi do osiągnięcia optymalnego celu. Zarządzanie taktyczne dotyczy głównie szczebla przedsiębiorstwa.

Fabryka Narzędzi Skrawających FREZER w Moskwie, po dokonaniu analizy uznała za celowe zastosowanie komputera w następujących zakresach:

— planowanie produkcji

— planowanie zatrudnienia i płac

— planowanie kosztów własnych.

Interesujące doświadczenia zdobyto przy wprowadzaniu ASZ w Fabryce Maszyn Górniczych w Doniecku [8], gdzie nomenklatura wyrobów waha się od 600 do 900 pozycji. Do produkcji używa się tam około 20 tysięcy rodzajów materiałów.

System ma następujące zadania do rozwiązania:

- opracowanie programów produkcyjnych przy optymalnym obciążeniu urządzeń
- opracowywanie harmonogramów obróbki części, zmierzających do zminimalizowania cyklów produkcyjnych
- czuwanie nad zapewnieniem terminowej kompletacji wyrobów.

Bardzo specyficznym zagadnieniem, któremu poświęca się wiele uwagi jest gospodarka materiałowa.

W Fabryce FREZER informacyjny system zautomatyzowanego zarządzania zaopatrzeniem materiałowo-technicznym [9] składa się z danych stałych, wykorzystywanych wielokrotnie oraz informacji zmiennych.

W Fabryce FREZER zorganizowano normatywną gospodarkę materiałową. Doświadczenia zdobyte przy normatywnej gospodarce materiałowej, zaprogramowanej na maszynie cyfrowej URAŁ 11, potwierdzają dużą efektywność wdrażania ASZ zaopatrzenia materiałowo-technicznego.

W Ministerstwie Obrabiarek i Instrumentów Przemysłowych ZSRR scentralizowano gospodarkę materiałową [10].

Interesujące doświadczenia zdobyto podczas opracowania i wdrażania systemów zarządzania produkcją chemiczną [11].

Produkcja chemiczna i związane z nią procesy ciągłe są bardzo złożone. Charakteryzuje je wielka liczba współzależnych parametrów. Ich matematyczny opis i modelowanie w celu optymalizacji całości procesu produkcyjnego często są niemożliwe. W takich złożonych systemach zarządzania z zastosowaniem komputera, całość procesu dzieli się na oddzielne, autonomiczne podsystemy. Dla każdego podsystemu formułuje się zadania odcinkowe i kryterium optymalizacji, skoordynowane z ogólnym kryterium.

Przykładem opracowania i przemysłowego wdrożenia takich systemów zarządzania są eksploatowane w Północnodonieckim Kombinacie Chemicznym systemy AWTODYSPETCZER, oparte o maszynę UM 1.

Wiele inspiracji do rozwiązań szeregu przemysłów można znaleźć w systemie optymalizacji procesu wstępnej rafinerii ropy [12]. Urządzenia do wstępnej rafinerii ropy są podstawowymi urządzeniami zakładów przerabiających ropę i one określają efektywność pracy zakładów. Proces charakteryzuje się zarówno wielkim tonażem surowca wprowadzonego, jak i wyrobu odbieranego. Dotychczas warunki pracy były określane subiektywnie przez operatora. Zastosowanie ASZ pozwala na optymalizację warunków przebiegu procesu. Nakłady na maszynę cyfrową zwrócą się w ciągu 2 lat.

Automatyzacja zarządzania operatywnego

Zarządzanie operatywne wiąże się głównie z czynnościami wykonawczymi. Znamy dokładnie cel, mamy określone zadania lub środki i dążymy do ścisłego wykonania zadań.

Zarządzanie operatywne ma zastosowanie w ramach przedsiębiorstwa i dotyczy jednego wydziału względnie nawet jednego urządzenia. Automatyzacja zarządzania operatywnego ma dwa nurty. Jeden z nich dotyczy spraw organizacyjno-optymalizacyjnych, drugi — zagadnień techniczno-technologicznych i wyraża się w sterowaniu procesami produkcyjnymi.

- Operatywne systemy organizacyjno-optymalizujące mają na celu takie bieżące organizowanie przebiegu działalności, które zapewniłoby optymalne wyniki w ramach istniejących warunków.

Zilustrujemy problem kilkoma przykładami.

W przedsiębiorstwie budowy maszyn opracowano i wdrożono system określania dróg przechodzenia detali i podzespołów wyrobów między wydziałami oraz stanowiskami pracy. System opracowano na maszynę MIŃSK 22 [13].

Innym przykładem jest system operatywnej wysyłki wydobytego węgla [14]. Grupa kopalń wraz z zarządem załadunku i transportu (ZZT) tworzy określony system. Występują dwa najbardziej rozpowszechnione sposoby zarządzania tym systemem.

W pierwszym przypadku kopalnie wysyłają węgiel bez gromadzenia w bunkrach, co zapewnia wysyłkę węgla lepszej jakości. W warunkach braku wagonów, węgiel gromadzi się w składach awaryjnych, lecz wówczas powstają straty i obniża się jakość węgla.

W drugim przypadku węgiel przechodzi przez bunkier.

W bunkrze węgiel też traci na jakości, ale mniej niż w składzie awaryjnym. Wykorzystanie bunkrów pozwala w ramach zarządu załadunków i transportu ZZT regulować podstawianie wagonów według priorytetów. W każdym momencie pokrywa się to zapotrzebowanie kopalni, którego niezaspokojenie z uwagi na wskaźniki produkcyjne może wywołać stosunkowo największe straty dla grupy kopalń jako całości.

Dzięki systemowi ZZT osiągnięto maksymalne korzyści z wykorzystania środków transportowych i z obniżenia do minimum strat załadunkowo-transportowych w procesie wydobycia węgla.

- Systemy sterowania procesami produkcyjnymi. Sterowanie procesem sprowadza się do automatyzacji współpracy pomiędzy obiektem i operatorem procesu technologicznego. Istnieje wyraźna zależność systemu zarządzania procesem od jego technologii. Z tego też względu podamy poniżej kilka przykładów w przekroju branż przemysłowych.

Żuczkow pisze [15], że podstawowym założeniem racjonalnej organizacji procesu wytwarzania w Charkowskiej Fabryce Elektromechanicznej jest jego rytmiczność, nadzorowana przez komputer.

W systemie sterowania maszyna wykonuje następujące czynności:

- przeprowadza statystyczną obróbkę pierwotnych informacji otrzymywanych od obiektu
- opracowuje uprzedzającą sygnalizację o odchyleniach od norm w pracy obiektu
- udziela instrukcji operatorowi w zakresie parametrów, co umożliwia ich optymalne ustawienie
- samoczynnie reguluje wykonawczymi organami w warunkach zautomatyzowanego zarządzania.

Wielu autorów pisze o automatyzacji wydziału walcowni i cięcia metalu.

Swerdlin i Trofimow [16] informują nas o systemie operatywno-dyspozytorskiego zarządzania wydziałami

prażenia przy produkcji kwasu siarkowego w ZSRR za pomocą maszyny UM 1.

Instytut Cybernetyki AN USSR i Rafineria Ropy w Riazaniu opracowały i przeprowadziły doświadczenia z systemem optymalizacji procesu wstępnej rafinerii ropy [12].

W radzieckim przemyśle materiałów budowlanych zastosowano system zautomatyzowanego sterowania pracą pieca tunelowego do wypalania cegły, a nawet procesem przygotowywania mieszanki betonowej.

W przemyśle spożywczym ZSRR szerokim frontem wdraża się automatyzację kontroli i regulacji poszczególnych parametrów (temperatura, ciśnienie, ilość, ...) jak również automaty wykonujące czynności konfekcjonowania, rozlewu, pakowania itp. [17].

W cukrowni Bratuszanskiej rozpoczynano od lokalnych systemów automatyki (dyfuzje, wirówki), a obecnie steruje się już wszystkimi podstawowymi oddziałami centralnie przez dyspozytora.

W przemyśle tekstylnym, w celu poprawy jakości i jednolitości przędzy, zastosowano automatyzację jej barwienia. W kombinacie tkacko-przędzalniczym w celu równomiernego podawania włókna wprowadzono automatyzację tego procesu.

Z przytoczonego, bardzo szerokiego wachlarza zastosowań elektronicznych maszyn cyfrowych do sterowania procesami technologicznymi wynika, że automatyzacja może mieć zastosowanie w każdej dziedzinie produkcji. Istotną cechą automatycznego sterowania procesami jest dokładność regulacji procesu.

5. PODMIOTOWY ASPEKT PROCEDURALNEJ STRONY SYSTEMÓW

Zgadzając się z tym, że celem tworzenia ASZ jest wsparcie decyzji gospodarczych dobrym zapleczem informacyjnym, uświadamiamy sobie konieczność zapewnienia prawidłowości działania tego zaplecza. Żądamy szybkich i poprawnych informacji, najpierw jednak trzeba stworzyć ku temu warunki, tj. system informacyjny. Tworzyć go musi człowiek w odpowiednich warunkach organizacyjnych z odpowiednimi urządzeniami technicznymi. ASZ wymaga właściwych procedur, które zwiążą pracę człowieka i maszyny.

Proceduralną stronę systemów zajmują się w ZSRR dwa rodzaje podmiotów. Z jednej strony są to instytucje, a z drugiej — ludzie.

● Warunki organizacyjno-instytucjonalne

W ZSRR istnieje Ministerstwo Technicznych Środków Automatyzacji i Systemów Zarządzania, które współpracuje z ministerstwami innych gałęzi przy opracowywaniu ASZ w różnych instytucjach i przedsiębiorstwach kraju. W 1968 r. opracowano w ten sposób i rozpoczęto wdrażanie 160 ASZP [1]. Ministerstwo nie tylko dostarcza sprzętu do automatyzacji, ale również kieruje pierwszym wszechzwiązkowym zjednoczeniem projektowo-montażowym do wdrażania techniki obliczeniowej AWTOMATIKA.

Zjednoczenie to zajmuje się projektowaniem i montowaniem maszyn cyfrowych oraz przygotowywaniem software-u [18], a w szczególności zautomatyzowanych systemów planowania i zarządzania przedsiębiorstwami, jak i systemów sterowania procesami technologicznymi. W prowadzonych pracach zjednoczenie to występuje jako główny wykonawca. Zawiera z kontrahentem umowę na całość prac aż do momentu oddania systemu. Do prac odcinkowych zjednoczenie nie może włączać podwykonawców. Istotną rolę przy tworzeniu systemów odgrywa współpraca ze zleceniodawcą.

Opracowywaniem naukowych podstaw organizacji zajmuje się szereg organizacji naukowo-badawczych. Centralny Naukowo-Badawczy i Projektowo-Technologiczny Instytut Organizacji i Techniki Zarządzania (CNIITZ) prowadzi badania obejmujące różne zagadnienia zarządzania przedsiębiorstwem.

Rysują się trzy różne kierunki prac prowadzonych obecnie nad automatyzacją organizacji i zarządzania w ZSRR [19]:

I) nastawiony na automatyzację organizacji produkcji i księgowości w przedsiębiorstwie

II) zmierzający do podniesienia efektywności zarządzania

III) podejmujący próbę syntetycznego ujęcia problemu automatyzacji zarządzania na różnych poziomach gospodarki.

Wyrazem planowego, perspektywicznego i systematycznego podejścia do całości zagadnienia są gałęziowe plany automatyzacji zarządzania tworzone przez ministerstwa. Ta droga zapewnia zlikwidowanie dostrzeganego opóźnienia w komputeryzacji przemysłu ZSRR w stosunku do krajów przodujących w tym zakresie.

● Pracownicy. Funkcjonowanie automatyzacji zarządzania uzależnione jest od przygotowania kadr znających się na naukowych metodach zarządzania i obeznanych z EPD.

Obecnie w ZSRR panuje przekonanie [14], że przygotowywanie kadr w zakresie nowych metod zarządzania powinno odbywać się dwoma kanałami, z których pierwszym jest nauczanie młodzieży w wyższych uczelniach, a drugim — dokształcanie specjalistów.

Za podstawową działalność zmierzającą w tym kierunku należałoby uważać organizację odpowiednich fakultetów i katedr w wyższych uczelniach republik.

6. PRZEDMIOTOWY ASPEKT PROCEDURALNEJ STRONY SYSTEMÓW

Przedmiotowy aspekt proceduralnej strony systemów zmierza do zapewnienia technicznych możliwości przetwarzania danych i uzyskiwania informacji. Wchodzą tu w grę:

— środki techniczne (hardware)
— oprogramowanie (software)
— ośrodki obliczeniowe
— sieć łączności.

● Środki techniczne występują w czterech podstawowych typach:

— maszyny cyfrowe
— małe urządzenia sterujące procesami
— urządzenia wejścia i wyjścia
— wszelkiego rodzaju urządzenia końcowe (środki do zbierania i przekazywania informacji).

Wszystkie wymienione typy urządzeń są produkowane seryjnie w ZSRR. Zwraca uwagę kompleksowość podejścia do tego zagadnienia.

● Oprogramowanie zapewnia właściwe — w sensie technologicznym — przetwarzanie danych.

Prace nad zapewnieniem prawidłowej technologii przetwarzania danych przebiegają w ZSRR dwupoziomowo. Wykonują je wyspecjalizowane instytuty dążące do opracowania typowych rozwiązań oraz programów lub realizują je branże względnie przedsiębiorstwa w celu zaspokojenia swoich specyficznych potrzeb.

Chafizow omawia sposób wyboru efektywnych metod programowania [20]. Autor wskazuje metodę pomiaru efektywności wariantów programowania ręcznego i maszynowego z zastosowaniem translatora. Wypowiedź ta zwraca uwagę na możliwość mechanizacji pracochłonnego projektowania.

W pokonywaniu trudności z oprogramowaniem maszyny pomocny jest w ZSRR ukraiński republikański fundusz algorytmów i programów (RFAP) przy Instytucie Cybernetyki Akademii Nauk w Kijowie. Na fundusz ten składają się algorytmy i programy do

[1]) Zautomatyzowanych Systemów Zarządzania Przedsiębiorstwami.

rozwiązywania zadań naukowych, inżynieryjno-technicznych i planowo-ekonomicznych. Ponadto wchodzą do niego instrukcje dotyczące zastosowań i wykorzystania tych programów.

● Sieć ośrodków obliczeniowych. Jeśli idzie o organizację ośrodków informacyjno-obliczeniowych (OIO), to podstawowa linia podziału przebiega pomiędzy OIO branżowymi a ich siecią ogólną, obsługującą różnych użytkowników. Niezależnie od tego podziału podstawowego, w ramach branżowych OIO można wyróżnić branżowe ośrodki przy jednostkach nadrzędnych, w przedsiębiorstwach lub zlokalizowane w miastach położonych centralnie względem grupy zakładów. Przy zakładaniu OIO powstaje problem optymalnego wyposażenia w środki techniczne OIO [21].

● Sieć łączności. W miarę pionowej integracji systemów zautomatyzowanego zarządzania gospodarką i pogłębiania się współpracy poziomej coraz większą rolę będzie odgrywać sieć do transmisji danych. Na to zagadnienie zwracają uwagę liczni autorzy [2].

7. METODY I FORMY TWORZENIA SYSTEMÓW

Metody i formy tworzenia systemów w ZSRR mają wiele cech specyficznych i mogą stanowić cenne wskazówki dla ludzi opracowujących systemy [2].

Daje się je podzielić na dwie podstawowe klasy. Pierwsze zmierzają do zapewnienia możliwie najdoskonalszej strony funkcjonalnej, czyli rozpatrywane są pod kątem widzenia użytkownika systemu. Drugie zmierzają do ułatwienia pracy twórcom systemu i do jej przyspieszenia.

Do metod doskonalących stronę funkcjonalną można zaliczyć:

1. Wyprzedzające konstruowanie modelu zarządzania rozpatrywanym obiektem

2. Ustalanie stopnia znaczenia oddzielnych ogniw zarządzania w aspekcie dynamicznym przedsiębiorstwa

3. Uprzednie doprowadzenie produkcyjnej struktury obiektu zarządzania do zgodności z wymaganiami naukowo-technicznymi procesu i systemu zarządzania

4. Ustalanie form i częstotliwości dokumentacji pierwotnej

5. Uzgodnienie administracyjnych, ekonomicznych i innych metod wpływania na wykonawców w toku procesów gospodarczych i produkcyjnych

6. Wybór racjonalnych etapów rozpracowywania ASZ.

Do metod doskonalących stronę proceduralną systemów można zaliczyć:

1. Organizację biegu strumienia dokumentacji między organem zarządzającym i obiektem zarządzanym

2. Ustalanie bezwzględnie obowiązującej sieci powiązań (harmonogramu) wszystkich elementów ASZ

3. Programowo właściwe zabezpieczanie procesu funkcjonowania ASZ w zgodzie z charakterem zadań zarządzania i specjalnościami struktury organizacyjnej aparatu zarządzania

4. Modelowanie projektowych rozwiązań w celu skrócenia terminów wdrożenia i opanowania ASZ.

Fundamentalnym zagadnieniem przy opracowywaniu systemu automatyzacji, który ma służyć przez wiele lat, jest właściwe i gruntowne jego przygotowanie.

Przy przygotowywaniu ASZ i jego wdrażaniu szczególnie zwraca się uwagę na poniższe zagadnienia:

1) analiza i poznanie przedsiębiorstwa
2) ocena prawidłowości organizacji i dokumentacji pierwotnej oraz ewentualne ich zmiany
3) formalizacja procedur postępowania
4) wybór komputera i innych środków technicznych
5) określenie sposobu oprogramowania
6) ocena efektywności systemu
7) zbieranie i kodowanie danych
8) poprawa jakości wyników
9) przyspieszenie prac wdrażanych
10) instytucjonalne zabezpieczenie wdrażania systemów.

Przy czynnościach przygotowawczych przeprowadza się badania zmierzające do poznania obiektu, oceny jego organizacji i dokumentacji. Poznanie charakterystyki przedsiębiorstwa sprowadza się do określenia specyfiki systemu informacyjnego, organizacyjnej struktury zarządzania, funkcjonalnych osobliwości istniejącego systemu zarządzania i podstawowych metod zarządzania. Obiektem poznania musi być zarówno przedsiębiorstwo jako całość, jak i jego strukturalne oddziały. Badania przeprowadza się na podstawie dokumentów i ich zawartości informacyjnej.

Kiriluk i Sirczenko [21] podają tablice, w które należałoby ujmować wyniki badań. Analiza charakterystyki i wskaźników umożliwia nakreślenie organizacyjnych przedsięwzięć, które należy zrealizować w okresie tworzenia ASZ.

Wdrażanie każdego systemu przeprowadza się etapowo. Rozróżnia się okres tworzenia systemu, jego doskonalenia oraz rozwoju. W ZSRR dominuje elastyczne podejście metodyczne przy tworzeniu systemów, które dostosowuje się do scentralizowanego systemu zarządzania względnie do zdecentralizowanych potrzeb podejmowania decyzji. Szczególnie efektywne jest automatyzowanie zarządzania większymi kompleksami produkcyjnymi — przedsiębiorstwami, kombinatami, zjednoczeniami, gałęziami itp. Inną istotną korzyścią metodologii postępowania jest tworzenie typowych rozwiązań programowych i unifikacja dokumentacji pierwotnej.

Centralny Naukowo-Badawczy i Projektowo-Technologiczny Instytut Organizacji i Techniki zarządzania prowadzi badania i opracowuje bazowe ASZP dla przemysłu maszynowego. Doświadczenia systemów bazowych mają być w przyszłości szeroko rozprzestrzenione na inne przedsiębiorstwa. Do bazowych ASZP tak dobrano przedsiębiorstwa, aby reprezentowały one wszystkie typy produkcji (masowa, seryjna, jednostkowa) i wszystkie rodzaje technologicznych przekrojów. Zwraca uwagę fakt, że w najbliższym czasie oczekuje się opracowania szeregu typowych systemów w różnych gałęziach produkcji.

Litwincew [23] podkreśla, że przy wdrażaniu ASZ istotne znaczenie ma analiza istniejącej dokumentacji i opracowanie nowych dokumentów w celu optymalizacji ich formy i zawartości. Praktyka bowiem wykazuje, że zawartość dokumentów nie zawsze jest w pełni wykorzystana.

Wraz ze wzrostem zapotrzebowania na informację o działalności przedsiębiorstwa, rośnie znaczenie danych ewidencyjno-statystycznych. Przy gromadzeniu tych danych bardzo istotne znaczenie ma jednolite podejście do ich kodowania [24].

8. WNIOSKI KOŃCOWE

Dalsze doskonalenie zarządzania gospodarką narodową w ogóle, a w przemyśle w szczególności jest możliwe tylko w ścisłym powiązaniu z automatyzacją, która stanowi nową technologię zarządzania.

Zrozumienie olbrzymich możliwości, jakie nowa technologia zarządzania stwarza dla organizmów gospodarczych, umożliwia kierowanie przemysłem w ZSRR na podstawach naukowych, wspartych badaniami, poznaniem i wykorzystaniem obiektywnych praw ekonomicznych oraz ostatnimi osiągnięciami nauki i techniki.

Wprowadzenie automatyzacji zarządzania jest w obecnych warunkach nie tylko podstawowym kierunkiem działania w celu dokonania skoku w poprawie dobrobytu mas pracujących. W ZSRR istnieje pełna świadomość potrzeby i celowości prowadzenia badań w zakresie metod zarządzania oraz konieczności wdrażania najnowocześniejszych osiągnięć do praktyki gospodarczej.

Obserwowany rozwój komputeryzacji ma swoje uzasadnienie w świadomości, że wyniki kierowania zależą od systemu informacyjnego.

Systemy zautomatyzowanego zarządzania wraz ze swoimi doskonałymi systemami informacji mają za zadanie zwalczenie ograniczeń poznawczych skomplikowanych procesów gospodarczych i produkcyjnych. ASZ mają zapewnić traf-

ność decyzji różnych szczebli zarządzania. Zwiększanie liczby działających ASZ powinno się przyczynić do wyeliminowania niesłusznych decyzji.

Niezależnie jednak od doceniania możliwości, a nawet konieczności wdrażania ASZ, obserwuje się nierówny rozwój przedmiotowych dziedzin zarządzania. Stosunkowo najwięcej systemów wdraża się w dziedzinie operatywnego sterowania i zarządzania. Mniej systemów funkcjonuje w dziedzinie zarządzania taktycznego. Rzadko spotykamy systemy zarządzania strategicznego. Stopień trudności przygotowywania systemu gra tu istotną rolę.

Oceniając ogólnie trzeba stwierdzić, że w ZSRR stworzono warunki do automatyzacji zarządzania. Docenia się pierwotne znaczenie momentu organizacyjnego i kompleksowości rozwoju składników systemu. Rozwija się produkcję sprzętu, przygotowuje się kadry i tworzy naukowe podstawy doskonalenia metod programowania.

Wzrost zainteresowania automatyzacją zarządzania, poparty środkami i ludźmi, zapewnia w perspektywie dokonanie w tej dziedzinie w ZSRR dużego kroku naprzód.

BIBLIOGRAFIA

[1] Koźmiński A. K., Zawiślak A. M. — „Bariera organizacji zarządzania". Miesięcznik Literacki, maj 1971, s. 93

[2] Modin A. — „Problemy razvitia awtomatizirovannych sistem upravlenija promyšlennostju". Voprosy ekonomiki, 1971, nr 3, s. 83

[3] Fedorenko N., Majminas E. — „K organizacii processov priniatija ekonomičeskich rešeni'. Voprosy ekonomiki, 1971, nr 3, s. 15

[4] Dygnikov E. G., Wolter B. V., Piggot S. G., Bernstein I. M. — „Raboty CNIIKA po avtomatizacii upravlenija obiektami chimičeskoj promyšlennosti". Mechanizacja i Avtomatizacija Proizvodstva" (M.iA.P.) 1970, nr 7, s. 1

[5] Kluev J. B. — „Algoritm perspektivnogo prognozirovanija energopotreblenija". Mechanizacija i Avtomatizacija Upravlenija. (M.iA.U.) 1970, nr 3, s. 1

[6] Żaganski A. I., Šeftel A. M. — „Optimizacija veličiny i razmeščenija strachovogo zapasa oborudovanija i zapčastej v energosisteme" M.iA.U. 1970, nr 2, s. 15

[7] Maksymov J. P., Kvartalnova V. V. — „Mechanizacija techniko-ekonomičeskogo planirovanija s primeneniem EVM". M.iA.P. 1970, nr 10, s. 42

[8] Liberman B. B., Amitan V. V. — „Avtomatizirovannaja sistema upravlenija mašinostroitelnym zavodom". M.iA.P. 1970, nr 3, s. 32

[9] Nizamowa S. N. — „Avtomatizirovannaja sistema upravlenija materialno-techničeskim snabženiem na predprijatii". M.iA.P. 1970, nr 8, s. 39

[10] V Ministerstve Promyšlennosti Stankov i Instrumentov SSSR M.iA.P. 1970, nr 5, s. 52

[11] Novochatnyj A. A., Ščerbašyn J. D., Galočkin M. M. — „Opyt razrabotki i vnedrenija sistem s EVM dla upravlenija dwumja chimičeskimi proizvodstvami". M.iA.U. 1970, nr 1, s. 26

[12] Kusovskij B. I., Dejneko P. S., Bakun G. M. — „Razrabotka i vnedrenije sistemy optimizacii processa pervičnoj pererabotki niefti". M.iA.U. 1970, nr 2, s. 26

[13] Lužankov I. N. — „Mechanizacija sostavlenija maršruta prochoždenija detalej i uzlov izdelija". M.iA.P. 1970, nr 9, s. 34

[14] Tkačenko I. S. — „Modelirowanie na EVM processa otgruzki ulga s šacht". M.iA.U. 1970, nr 1, s. 3

[15] Žučkow M. I. — „Soveršenstvovanie organizacii proizvodstvennych processov pri izgotovlenii stancji upravlenija". M.iA.U. 1970, nr 4, s. 11

[16] Swerdlin W. M., Trofimow W. I. — „Sistema operativno-dispečerskogo upravlenija obžigovym cechom sernokislotnogo proizvodstva". M.iA.P. 1970, nr 5, s. 49

[17] Pietrov I. K., Komlev A. I. — Avtomatizacija technologičeskich processov piščevoj promyšlennosti". M.iA.P. 1970, nr 7, s. 6

[18] Balykin W. — „O pervom Wsiesojuznom projektno-montažnom objedienii po vnedrieniju wyčislitelnoj techniki" — „Pribory i sistiemy upravlenija". 1970, nr 6, s. 54

[19] Informacje z zagranicy: „Co przynosi elektroniczne przetwarzanie danych?", Zjednoczenie Przemysłu Automatyki i Aparatury Pomiarowej MERATECH, kwiecień 1969 r.

[20] Chafizow F. I. — „Wybor effektivnych metodow programirovanija pri obrabotke informacii na EVM". M.iA.P. 1970, nr 8, s. 48

[21] Kolinko A. I., Kiriluk N. I. — „Vybor optimalnogo naraščivanija vyčislitelnych sriedstv vyčislitelnogo centra obrabotyvajuščego informaciju dla upravlenija predprijatiem". M.iA.U. 1970, nr 2, s. 20

[22] Kiriluk P. I., Sirczenko A. G. — „Izocenije i analiz charakteristik priedprijatija pri razrabotke projektov ASUP". M.iA.U. nr 2, s. 37

[23] Litvincew A. V. — „Metod statističeskogo analiza ispolzovanija učotnych dokumentov po forme i soderžaniju pri vnedreniju ASUP". M.iA.U. 1970, nr 4, s. 19

[24] Pinczuk R. N., Gabelko V. I., Voronkov O. G. — „Odin podchod k kodirovaniju učetno-statističeskoj informacii". M.iA.U. 1970, nr 3, s. 13.

PRZEGLĄD WYDAWNICTW

ELEKTRONICZNA TECHNIKA OBLICZENIOWA — NOWOŚCI

ETO-NOWOŚCI kwartalnik wydawany przez BOITE Instytutu Maszyn Matematycznych jest poświęcony aktualnym informacjom o stanie, postępach i perspektywach rozwoju techniki i technologii elektronicznych maszyn cyfrowych. Aktualnie ukazały się dwa zeszyty:

nr 1/71 **ETO-NOWOŚCI** zawierający następujące artykuły —

- Zbigniew Świątkowski — „IV Kongres Mikroelektroniczny i Międzynarodowe Targi Elektroniczne 70"
- Jan Reluga — „Nowoczesna aparatura technologiczna do produkcji dwu- i wielowarstwowych płytek z obwodami drukowanymi"
- Jerzy Ekner, Mirosław Mika — „Wielowarstwowe płytki drukowane. Cz. II. Uwagi o technologii"

Krótkie informacje: „Z Kraju", „Ze Świata"

nr 2/71 **ETO-NOWOŚCI** zawierający następujące artykuły —

- Piotr Perkowski — „Przegląd zagadnień symulacji procesów dyskretnych"
- Włodzimierz Mardal — „Nowe opracowania i tendencje w dziedzinie bardzo małych maszyn cyfrowych (minikomputerów)"
- Antoni Kwiatkowski — „Stan obecny i perspektywy rozwoju pamięci taśmowych i dyskowych"
- Andrzej Siuda — „Konstruowanie pulpitów sterowniczych"
- Stefan Parvi — „Nowe rozwiązania konstrukcyjne i sposoby wykonania głowic pamięci kinetycznych maszyn cyfrowych"

Krótkie informacje: „Z Kraju", „Ze Świata"

„Komunikat Międzynarodowego Stowarzyszenia Cybernetycznego"

W. W.

Bibliografia książek polskich z dziedziny informatyki

- **Systemy programowania maszyny cyfrowej ODRA 1204 ALGOL 1204. 1204 — VIII — 2** — JERZYKIEWICZ K., SZCZEPKOWICZ J. Wrocławskie Zakłady Elektroniczne ELWRO. Wrocław 1970, ss. 105.

Język ALGOL 1204. Procedury i zmienne standardowe. Procedury specjalne. Obsługa translatorów i programów. System obsługi i nadzoru MASON. Dodatki: 1. Wykaz zleceń ALGOLu 1204. 2. Standardowe znaczenie klawiszy nry 0, 1, 2, 3 i 4 w ALGOLu 1204. 3. Skorowidz nazw, sygnałów i symboli. Materiały szkoleniowe dla programistów. Od czytelnika wymaga się znajomości języka algorytmicznego ALGOL 60.

- **Zasady działania maszyn cyfrowych i programowania** — ORMISKI A .— Wyd. 2 uzupełnione. Wyd. Akademii Górniczo-Hutniczej w Krakowie, Kraków 1970, ss. 179, cena zł 13,— (skrypt).

Wydanie z 1 roku 1969 omówiono w czasopiśmie „Maszyny Matematyczne" 1969, nr 9, III str. okładki. W wydaniu 2 umieszczono nowy rozdział 5 pt. „Automatyzacja programowania". Skrypt napisany jest na podstawie wykładów prowadzonych na Wydziale Elektrotechniki Górniczej i Hutniczej Wieczorowego Studium Inżynierskiego AGH dla specjalności Automatyka.

- **Elektroniczna maszyna cyfrowa ODRA 1304.** Dokumentacja techniczno-ruchowa. Urządzenia. Jednostka centralna. T. 6. Pakiety, schematy montażowe i ideowe. Wyd. ELWRO, Wrocław 1970, nlb.

Schematy są przeznaczone dla użytkowników ODRY 1304.

ZBIGNIEW ŁADOŚ
Centrum Elektroniczne NBP, Warszawa
SŁAWOMIR TRAUTMAN
ZOWAR

681.322.06.004.13

Z praktyki programowania systemów API

Publikacja jest stenogramem z dyskusji dwóch osób będących kierownikami zespołów projektująco-programujących w ośrodkach obliczeniowych Narodowego Banku Polskiego oraz w ZOWAR. Ma ona na celu zwrócenie uwagi, szczególnie użytkowników, na problem niezawodności oprogramowania. Wskazuje również na konieczność unormowania wielu spraw związanych z upowszechnieniem stosowania komputerów.

Pytanie 1

Jeżeli programista opracowuje wyjątkowo duży i logicznie bardzo skomplikowany program dotyczący API, czy w praktyce może on w 100% sprawdzić na komputerze wszystkie kombinacje działania tego programu?

Odpowiedź A

W niektórych skomplikowanych i dużych programach dotyczących API (np. kontroli logicznej i rachunkowej danych wejściowych, obliczeń podatkowych w powiązaniu z różnymi przepisami płacowymi itp.) istnieje teoretycznie wiele tysięcy kombinacji logicznych do sprawdzenia. Szacuje się, że takich programów jest na ogół niedużo (około 10%), ale w każdym systemie stanowią one największy problem. Do opracowania tego rodzaju programu powinno się wyznaczyć doświadczonego programistę, który potrafi logicznie powiązać dużą liczbę różnych rozgałęzień w programie.

Przy założeniu, że osoba zlecająca opracowanie tego programu dobrze zna z praktyki temat, zaś dla programisty temat również nie jest całkiem obcy, wówczas istnieje szansa wyodrębnienia prawie wszystkich występujących w praktyce powiązań logicznych i sprawdzenia na komputerze około 30—50% ogólnej ilości teoretycznych wariantów powiązań logicznych w programie. Oczywiście wszystkie zwykłe rozgałęzienia w programie muszą być raz lub kilka razy sprawdzane. Marginesem ryzyka około 50—70% są powstałe nie sprawdzane, niezwykle liczne wzajemne logiczne powiązania pomiędzy różnorodnymi fragmentami programu. Sprawdzenie wszystkich teoretycznych wariantów związków logicznych w programie byłoby nieopłacalne, gdyż przypadki takie w praktycznym przetwarzaniu na komputerze prawdopodobnie nigdy by się nie zdarzyły. Opracowanie i sprawdzenie na maszynie takiego programu jest wielokrotnie trudniejsze niż innych przeciętnych programów w przetwarzaniu danych, których jest większość (około 90%).

Gdy jednakże dla ośrodka programującego temat do programowania jest nowy i nietypowy, szczególnie ważną sprawą jest opracowanie przy pomocy doświadczonego praktyka — specjalisty od przyszłego użytkownika komputera, bardzo wszechstronnych, fikcyjnych danych do testowania takiego programu. Niekiedy jest to zadanie niewiele mniej pracochłonne od opracowania samego programu. Zasadnicza bowiem trudność polega na szczegółowym wyodrębnieniu, które kombinacje logiczne różnych rozgałęzień w programie najprawdopodobniej w praktyce wystąpią — nawet, gdy są to przypadki unikalne — a które najprawdopodobniej nie wystąpią.

W związku z tym przywiązuje się dużą wagę do doboru osób, jakie wyznaczy przyszły użytkownik komputera do współpracy z projektantami i programistami ośrodka programującego. Niekiedy wybór ten determinuje wszechstronność sprawdzenia i niezawodność działania programów użytkowych.

Odpowiedź B

Jak już wspomniano, nieopłacalne byłoby sprawdzanie na komputerze wszystkich teoretycznie możliwych kombinacji logicznych, wyjątkowo dużego i skomplikowanego programu.

Niemniej jednak wystąpią w praktyce takie programy, które są zarówno skomplikowane jak i niezwykle ważne „strategicznie" wśród wielu programów obsługiwanych w tym temacie na komputerze, np.: programy aktualizacji.

Trzeba koniecznie w tych programach w 100% sprawdzić logikę programu. Błędy logiczne w takim „kluczowym" programie mogą spowodować bardzo duże straty u użytkownika maszyny, np. wystąpił w praktyce przypadek straty rzędu 0,5 mln zł wskutek logicznie błędnego działania programu aktualizacji.

Wszelkie nietypowe przebiegi takiego programu, dla których nie można z góry przewidzieć odpowiednich rozwiązań, powinny spowodować przerwanie przetwarzania oraz podanie pisemnej informacji przez maszynę o typie powstałego błędu wraz z podaniem przybliżonego miejsca powstania błędu w programie. Sprawa wymaga przebadania przez osoby, które dobrze orientują się w działaniu programu. Mniejszym „złem" jest przerwanie przetwarzania z błędami, niekiedy trudnymi do szybkiego wykrycia. Czasami, jeśli program mógłby być logicznie zbyt trudny do sprawdzenia, warto zastanowić się, czy nie lepiej podzielić go na dwa mniejsze programy.

Pytanie 2

Jeżeli w wyjątkowo dużym i skomplikowanym logicznie programie programista nie może sprawdzić wszystkich kombinacji logicznych działania tego programu, czy ośrodek projektujący i programujący powinien o tym zawiadomić użytkownika i w jakiej formie winien to uczynić?

Odpowiedź A

Użytkownik programów, który już z doświadczenia zna problemy dotyczące EPD, przeważnie wie o tym, że w niektórych skomplikowanych programach nie opłaca się testować wszystkich teoretycznych kombinacji. W większości przypadków lepiej jednak pisemnie poinformować użytkownika, jaki jest stan faktyczny, tzn. które programy są najbardziej skom-

plikowane i nie mogły być w pełni przetestowane na maszynie. Użytkownik może wówczas, jeśli uzna to za potrzebne, zlecić dodatkowe przebadanie działania takiego programu, podając szczegółowe informacje, jakie przypadki powinny być przebadane.

Odpowiedź B

Wskazane jest podanie takiej informacji pisemnej dla użytkownika, aby nie stawiał on w przyszłości zarzutów, że programy nie zostały dotestowane na maszynie.

Pytanie 3

Jeżeli w programie wyjątkowo dużym i skomplikowanym, mimo bardzo częstego, bezbłędnego przetwarzania użytkowego na komputerze, wystąpi nagle po roku lub po kilku latach błąd w programie, czy należy to ocenić jako:

a) przypadek normalny?
b) przypadek wyjątkowy?
c) przypadek niedopuszczalny?

Odpowiedź A

Jest to przypadek wyjątkowy, który w praktyce zdarza się bardzo rzadko. Niekiedy skutki takiego błędu mogą być fatalne w przetwarzaniu na komputerze. Istnieje tu pewna analogia do przejmowania standardowego programowania komputerów, jakie dostarcza producent maszyny dla ośrodka. Nawet czołowi producenci komputerów w świecie nie są w stanie zapewnić bezbłędnego w 100% oprogramowania standardowego komputerów. Każdy ośrodek wykorzystuje otrzymane programy przeważnie w granicach, jakie zostały sprawdzone, równocześnie badając, jakie nowe funkcje tych programów można w sposób bezbłędny stosować.

Odpowiedź B

Są to unikalne błędy, których nikt nie był w stanie przewidzieć. Nie zostały one też wykryte przez użytkownika przy kontroli związanej z odbiorem programów. Nie dotyczy to oczywiście 90% programów małych i średnich lub nawet dużych, ale niezbyt skomplikowanych, w których można w zasadzie dokładnie sprawdzić wszystkie fragmenty.

Pytanie 4

Jak długo programista jest odpowiedzialny za ewentualne braki w programie? Jak długo ośrodek odpowiada za braki w programie w stosunku do użytkownika tych programów?

Odpowiedź A

Programista jest służbowo odpowiedzialny za poprawność działania swego programu tak długo, jak długo jest zatrudniony w tym ośrodku. Ewentualnie ujawnione błędy mogą mieć wpływ np. na premiowanie, awansowanie, a w skrajnym przypadku, przy częstych błędach mogą doprowadzić do konieczności przekwalifikowania pracownika do innego zawodu. Jednakże gdy programista jest zajęty innym pilnym programem nie zawsze może od razu zająć się poprawieniem swego starego programu.

Ośrodek jest obecnie bezterminowo odpowiedzialny wobec użytkownika (odbiorcy programów) za błędy w programach. Brak przepisów, które umożliwiałyby odpowiednie uregulowanie tej sprawy, jest źródłem nieporozumień pomiędzy ośrodkami a niektórymi użytkownikami. Rozwiązanie obecne jest nieżyciowe, albowiem w ośrodkach programujących przeważnie istnieje płynność kadr, zaś dla innego programisty jest to bardzo pracochłonna czynność znalezienie błędu w cudzym programie.

Niekiedy trzeba nawet zrezygnować z poszukiwań i opracowywać program od nowa. Również dla autora programu badanie „starego" programu jest bardziej pracochłonną czynnością, gdyż na ogół nie pamięta on szczegółów i musi studiować dokumentację programu. Brak jakichkolwiek ustaleń w tej sprawie powoduje, iż niektórzy użytkownicy dokonują odbioru programów dość powierzchownie, nie bardzo zdając sobie sprawę z trudności, jakie mogą u nich powstać w razie wystąpienia takich błędów.

Odpowiedź B

Użytkownik programów winien dokonać bardzo wnikliwego i wszechstronnego sprawdzenia programów przed ich odbiorem. W tym celu powinien przygotować własne, fikcyjne dane do prób na komputerze wykonywanych przed odbiorem programów. Nie powinno się też zbytnio skracać okresu dublowania pracy, tj. wykonywania jej metodą tradycyjną i za pomocą komputera. Użytkownik powinien też korzystać z komputera i jego programów w zakresie przez siebie sprawdzonym. W przypadkach bardziej skomplikowanych systemów lub podsystemów API, odbiór przez użytkownika programów w krótkim czasie nie jest możliwy i dlatego jest wskazane wprowadzenie okresu gwarancyjnego dla programów, który trwałby średnio około 6 miesięcy. Byłaby to analogia do sprzedaży skomplikowanych maszyn, które producent wprawdzie testuje, ale dalszy ciąg sprawdzania odbywa się u odbiorcy w okresie gwarancyjnym.

Wyjątkowo, dla programów o bardzo zróżnicowanych okresach przetwarzania, można by przykładowo wprowadzić okresy gwarancyjne, dla programów przetwarzanych w cyklu:

- codziennym — gwarancja 3-miesięczna
- miesięcznym — gwarancja 12-miesięczna
- rocznym — gwarancja 3-letnia.

W okresie gwarancyjnym, ośrodek programowy powinien być zobowiązany do bezpłatnego usuwania błędów w programach. Po tym okresie usuwanie błędów byłoby przez ośrodek dokonywane odpłatnie. Wprowadzenie okresu gwarancyjnego z pewnością zmobilizowałoby użytkowników do wnikliwszego sprawdzania odbieranych programów.

Pytanie 5

Zbiór programów komputera dla określonego tematu zawiera bardzo szczegółowe algorytmy postępowania, które przeważnie w naszych warunkach muszą być permanentnie udoskonalane i dostosowywane do zmieniających się potrzeb użytkownika. Jak ma z tym radzić sobie użytkownik komputera? Jakie ryzyko jest z tym związane?

Odpowiedź A

Na wstępie trzeba wyjaśnić, iż istnieją pewne granice braku stabilności w poszczególnych rozwiązaniach. Można je zaobserwować jeszcze przed rozpoczęciem opracowywania programów dla komputera. Określone zadania nie kwalifikują się do obsługi za pomocą komputera, dopóki ta dziedzina pracy u użytkownika nie zostanie odpowiednio uporządkowana, tj. uregulowana przepisami lub unormowana w inny sposób. Aczkolwiek doświadczeni projektanci API potrafią zaprojektować systemy, w których różne elementy mogą być zmienne, niemniej jednak pewne minimum stabilności jest konieczne.

Większość użytkowników komputerów musi z reguły liczyć się z koniecznością tzw. „konserwacji" programów, które bądź to będą zlecali ośrodkowi programującemu, bądź też będą je wykonywać przy pomocy własnych programistów lub w formie prac zleconych.

Odpowiedź B

W praktyce zauważa się, iż potrzeby wielu użytkowników komputerów powodują konieczność prawie sta-

łej pracy nad przerabianiem programów. Są jednak pewne granice w dokonywaniu tych przeróbek, po których nie można już programu zmieniać i trzeba go opracowywać od nowa. Granice te wyznacza:

a) sposób opracowania programu przez programistę. Wymaga się obecnie od każdego programisty, aby program był opracowany w sposób ułatwiający w przyszłości przeróbki, a więc nie liniowo, lecz w formie zwartych podprogramów, przejrzyście, z objaśnieniami i odpowiednią dokumentacją. Brak tych umiejętności całkowicie dyskwalifikuje programistę, albowiem nie można przyjąć programu, w którym zbyt trudno byłoby wnieść zmiany,

b) rozmiary zmian żądane przez użytkownika. Zbyt duża liczba zmian powoduje, iż opłaci się program opracowywać od nowa,

c) liczba już dawniej dokonanych zmian w programie.

Dla oszczędności proste zmiany w programie dokonuje się metodą tzw. łatwizny, tj. wplata się fragmentarycznie nowe rozkazy bez ponownego kompilowania, czyli tłumaczenia programu na komputerze za pomocą „języka", w którym program został pierwotnie napisany. W rezultacie po kilku latach w ośrodku dysponującym komputerem II generacji niemal wszystkie programy są tak poprawione. Przy komputerach III generacji możliwości tzw. łatwizny programów są bardziej ograniczone, ale częściowo również możliwe.

Po wielu zmianach w programie wprowadzanych opisywaną metodą program staje się nieczytelny nawet dla autora, a ponowna rekonstrukcja programu w celu przetłumaczenia na komputerze jest tak pracochłonna i ryzykowna, że niekiedy programiści wolą program opracować od nowa, tym bardziej, że równocześnie wymaga on na ogół generalnej reformy. Oczywiście przy takim opracowywaniu programu można wykorzystać pewne fragmenty starego programu.

Użytkownicy skomplikowanych systemów API powinni zdawać sobie sprawę z ryzyka w przetwarzaniu danych, zgłaszając życzenia zmiany programów. W logicznie skomplikowanym programie każda niekompletnie wniesiona zmiana kilkudziesięciu lub kilkuset rozkazów, na ogólną liczbę kilku tysięcy rozkazów tego programu, może spowodować błąd w zupełnie innym miejscu programu, które było uprzednio logicznie powiązane ze zmienianym fragmentem, czego zaś poprawiający mógł nie zauważyć.

Mimo więc, że wnoszona zmiana zostanie dokładnie przetestowana na komputerze przed użytkowanym przetwarzaniem, to opisywany błąd może być zupełnie wykryty, przy czym może on się ujawnić dopiero po pewnym okresie przetwarzania, w zupełnie nieoczekiwanym momencie. Przypadki takie już w praktyce występowały, np. przy zmianach w programie dotyczących zasad miesięcznych obliczeń płacowych, przy czym błąd programu został wykryty dopiero w listach płac przy obliczeniach dokonywanych raz w roku, dotyczących wysługi lat.

Przetestowanie na komputerze w 100% całego programu po wniesieniu każdej zmiany jest w praktyce niemożliwe, albowiem trwałoby zbyt długo i byłoby zbyt kosztowne. Zmiany w programach są z reguły niezwykle pilne i nie ma czasu na wszechstronne sprawdzenie całego programu. Oczywiście nie każda zmiana w programie jest tak ryzykowna. Doświadczenie wykazuje jednak, że po wprowadzeniu zmian do tego rodzaju programów jest konieczna wzmożona czujność podczas przetwarzania użytkowego w komputerze dotycząca programistów (specjalne dyżury programistów) i użytkownika (dodatkowa kontrola otrzymanych tabulogramów) przez dłuższy okres czasu.

Pytanie 6

Czy projektant jest odpowiedzialny za 100% kompletny zbiór informacji w założeniach przekazywanych programiście do opracowania programu?

Odpowiedź A

W praktyce można wyodrębnić trzy grupy projektantów:
a) projektanci do badań i analizy problemu, którzy określają potrzeby i projektują systemy informatyczne, zatrudnieni przeważnie w jednostkach prowadzących prace badawczo-rozwojowe,

b) projektanci do realizacji systemów elektronicznego przetwarzania danych, z reguły zatrudnieni w komórkach programujących komputery, opracowujący technologiczne założenia realizacji API, m.in. szczegółowe założenia dla indywidualnych programów, przekazywane programistom,

c) projektanci-programiści, realizujący mniejsze systemy API kompleksowo, tj. od projektu technologicznego aż do opracowania programu włącznie.

Od projektantów ad b) i c) wymaga się udokumentowanych, bardzo dobrych praktycznych umiejętności programowania komputerów. Każdy z tych projektantów zdaje sobie sprawę, że dowolny program, który jest przewidywany w jego projekcie technologicznym, może mu być zlecony do programowania. W związku z tym projektanci wiedzą dobrze, jakie informacje szczegółowe są potrzebne do programowania. Przy realizacji dużych systemów API zarysowuje się wyraźnie podział pracy pomiędzy projektantów ad b) i programistów. Niemniej jednak aby nie wyjść z wprawy projektanci ad b) od czasu do czasu również opracowują programy.

Przy realizacji mniejszych systemów API podział taki nie jest potrzebny i projektanci są zarazem programistami.

We wszystkich tych przypadkach wydaje się, że projektanci ad a) są odpowiedzialni za kompletność przeprowadzonych badań i zgłoszonych potrzeb i projektów, zaś projektanci ad b) i ad c) za kompletność szczegółów w zrealizowanych systemach API.

Odpowiedź B

W niektórych, dużych projektach API istnieją wycinki podtematów, np. pewna grupa typowych zestawień statystycznych, dla których projektant określa ramowe założenia, natomiast można programiście powierzyć obowiązek zaprojektowania szczegółowych założeń. Założenia te powinny być jednak skontrolowane i zatwierdzone przez projektanta odpowiedzialnego za realizację systemu API. Tego rodzaju praktyka umożliwia szkolenie programistów w pracach projektowych dla realizacji SEPD.

Pytanie 7

Czy w związku z powyższymi wyjaśnieniami istnieje duże ryzyko dla użytkownika przy rozpoczynaniu przetwarzania danych na komputerze?

Odpowiedź A

Użytkownicy większych systemów lub podsystemów API na komputerach dzielą się na dwie grupy, tj. na takich, którzy już orientują się w tej dziedzinie (są to zarówno doświadczeni użytkownicy, jak i zupełnie nowi, ale zapoznani z doświadczeniami innych użytkowników) oraz na zupełnie nie zorientowanych. Użytkownicy pierwszej grupy są świadomi ryzyka każdego nowego działania w tej dziedzinie (nowe programy, zmiany programów). Wiedzą oni, że jest z tym związane ryzyko, nieuniknione wszędzie tam, gdzie ma się do czynienia z postępem technicznym. Użytkownicy drugiej grupy wymagają specjalnie troskliwej obsługi w ośrodkach API do czasu, dopóki nie nabiorą odpowiednich doświadczeń.

Odpowiedź B

Świadomość ryzyka związanego z przetwarzaniem masowych danych na komputerze, za pomocą dużej liczby programów, w tym również skomplikowanych i

dużych programów, mobilizuje pracowników użytkownika do różnorodnych poczynań zmniejszających w rezultacie liczbę błędów w przetwarzaniu. Do takich poczynań należą m.in. staranne zgłoszenie wszystkich swych potrzeb, odpowiednie przygotowania organizacyjne poprzedzające wprowadzenie obsługi na komputerze (szkolenie, instrukcje itp.), opracowanie wszechstronnych, fikcyjnych danych modelowych służących do sprawdzenia i odbioru programów, wniklika kontrola tabulogramów podczas dublowania itd.

Aby system API prawidłowo funkcjonował, wymaga się dobrej roboty zarówno od pracowników ośrodka projektującego i programującego, jak również od pracowników przyszłego użytkownika komputera. Całkowicie mylny jest pogląd niektórych użytkowników, że automatyzacja ich pracy przy użyciu komputerów powinna być przeprowadzona „automatycznie" bez dużego udziału z ich strony w pracach organizacyjno-przygotowawczych.

JOWITA KONCEWICZ
Instytut Maszyn Matematycznych
Warszawa

681.322.06

SNOBOL - język do przetwarzania napisów

Przedstawiono język do przetwarzania napisów SNOBOL 3. W szczególności omówiono postać instrukcji, reguły tworzenia i przekształcania napisów, określenia skoku, operacje arytmetyczne, zmienne napisowe, określenie pośrednie napisu oraz funkcje standardowe i funkcje definiowane w programie.

Tylko bardzo wczesne zastosowania maszyn cyfrowych dotyczyły prawie wyłącznie obliczeń naukowych i inżynieryjnych oraz przetwarzania danych handlowych. Podczas rozwiązywania zagadnień z tych dziedzin dane wejściowe dla programów traktowano jako odpowiednio zakodowane liczby, na których maszyna wykonywała określone programem operacje matematyczne.

Jednakże w miarę tworzenia coraz to nowych języków programowania pojawiała się potrzeba układania translatorów dla tych języków — programów tłumaczących z języka programowania na język maszyny cyfrowej. Powstał wówczas nowy typ danych: teksty tłumaczonych programów, tj. napisy składające się z liter, cyfr i innych symboli — przestankowych (np. , . ;); operacyjnych (np. + — /) itp.

Początkowo niewiele osób miało do czynienia z tym nowym typem danych; byli nimi twórcy translatorów, którzy najczęściej pisali je w językach ukierunkowanych maszynowo — niezbyt do tego celu wygodnych, ale pozwalających programować oszczędnie. Wkrótce jednak zastosowania maszyn cyfrowych rozszerzono do takich zagadnień, jak próby tłumaczenia tekstów w językach naturalnych, badania porównawcze utworów literackich, próby określania autorstwa, symboliczne przekształcanie formuł matematycznych [1]), automatyczne tworzenie bibliografii i wyszukiwanie informacji, systemy konwersacyjne pracy z maszyną cyfrową. We wszystkich tego typu zagadnieniach danymi wejściowymi dla maszyny cyfrowej są ciągi symboli zwane napisami, na których wykonywane operacje powodują powstawanie nowych napisów. Poczęto wówczas tworzyć języki programowania przeznaczone specjalnie do przetwarzania napisów: LISP 1,5, COMIT, IPL, FORMAC, SNOBOL, TRAC, MATHLAB i wiele innych.

Pierwsze trzy z tych języków zostały opisane dość pobieżnie w [4], tym razem spróbujemy zaznajomić się bardziej szczegółowo z językiem SNOBOL, który ma stosunkowo prostą budowę, a przy tym duże możliwości zastosowań; doczekał się też wielu realizacji i kolejnych rozszerzeń.

Po około pięciu latach stosowania języka COMIT, kiedy zdawano już sobie sprawę z jego najważniejszych niedostatków, powstała w roku 1962 pierwsza wersja języka wzorowanego na nim, nazwanego SNOBOL. Zrealizowano go dla maszyny cyfrowej IBM 7090 w instytucji BELL TELEPHONE LABORATORIES.

Pierwszy opis tego języka [2] ukazał się w roku 1964. Wkrótce potem powstały nowe, rozszerzone wersje, z których najbardziej znana jest druga wersja SNOBOL-u 3, opisana szczegółowo w [1]. SNOBOL 3 zrealizowano m.in. dla maszyn cyfrowych: IBM 7090/94, IBM 7044, IBM 1620, CDC 3600, RCA 601/604, SDS 930/940, IBM System/360.

W roku 1967 powstał SNOBOL 4 zrealizowany dla maszyny IBM System/360, jednakże istnieją jedynie niepublikowane opisy tego języka.

W tym artykule zajmiemy się językiem SNOBOL 3, o którym pojawiło się stosunkowo dużo publikacji. Nie ukazał się jednak dotychczas żaden sformalizowany opis (typu raport), a we wszystkich publikacjach opisy są zależne od realizacji maszynowej, a więc od translatora, mimo że sam język jest od maszyny niezależny.

Napisy i operacje na nich wykonywane

Zanim przejdziemy do prezentacji języka SNOBOL, omówimy kilka podstawowych pojęć z dziedziny przetwarzania napisów.

Napis jest to dowolny ciąg symboli z alfabetu ustalonego dla danego języka. Początek i koniec napisu

[1]) Por. np. A. W. Mostowski — „Maszyny matematyczne w logice i algebrze", cz. I, „Maszyny Matematyczne" nr 5/1969.

przyjęto oznaczać symbolem apostrofu. Przykłady napisów:

"123ABCD456DEF"

"CZARNY KOT"

"I + J — 4 (X + Y)"

Każdemu napisowi można przyporządkować liczbę całkowitą zwaną *długością napisu*, np. długością pierwszego napisu w przykładzie jest liczba 13 (apostrofy nie należą do napisu), długością drugiego jest liczba 10 (odstęp wewnątrz napisu jest symbolem odstępu i jest traktowany tak samo, jak każdy symbol znajdujący się między apostrofami). Napis o długości zero nazywamy napisem pustym.

Na napisach można wykonywać dwie podstawowe operacje: *dołączanie* (ang. *juxtaposition)* oraz *połączenie*, zwane częściej konkatenacją (ang. *cocatenation)*. Pierwsza z nich polega na umieszczaniu napisów obok siebie, przy czym stykają się one, lecz pozostają w dalszym ciągu odrębnymi napisami; dołączenie napisu "A" do napisu "B" zapisujemy jako "A" "E". Druga operacja powoduje „zlepienie" łączonych napisów w jeden napis; konkatenacja napisu "BOR" z napisem "OWCA" da w wyniku "BOROWCA". Konkatenacja dowolnej liczby napisów jest zawsze jednym napisem, dołączenie natomiast jest ciągiem napisów stykających się ze sobą bezpośrednio, które można traktować jako jeden napis.

Oprócz tych dwóch podstawowych operacji istnieje wiele innych, które można podzielić na trzy zasadnicze grupy:

(1) *uzgadnianie z wzorcem* (ang. *pattern matching),*
(2) *rozłączanie* (ang. *decomposition)* oraz
(3) *zmienianie* (ang. *alteration).*

Uzgadnianie z wzorcem polega na sprawdzaniu, z jakich symboli składa się dany napis, np. można badać, czy w napisie "ALFA" drugim symbolem jest litera "L" lub, czy znajduje się w nim symbol "+". Operacje rozłączania powodują utworzenie kilku nowych napisów z jednego napisu, wreszcie zmienianie polega na zastępowaniu pewnych symboli w napisie innymi symbolami lub ciągami symboli. Napis powstały w wyniku zmieniania może mieć inną długość niż pierwotny, ponieważ można na miejscu jednego symbolu umieścić kilka innych lub kilka symboli sąsiednich zastąpić jednym.

Operacje ze wszystkich wymienionych grup można w dowolnie wybranym porządku stosować kilkakrotnie do jednego napisu. Na przykład:

1. Zastępując w napisie "KRAKUS" literę "U" napisem "OWIA" oraz literę "S" literą "K" otrzymamy napis "KRAKOWIAK"

2. Rozłączając napis "MARYSIA" na napisy "MARY" oraz "SIA" i zastępując pierwszy z nich literą "I" otrzymany w wyniku konkatenacji tych dwóch napisów napis "ISIA".

Po krótkim wprowadzeniu przejdziemy teraz do opisu języka SNOBOL.

Postać instrukcji w SNOBOL-u

Program w SNOBOL-u jest to ciąg *instrukcji.* Instrukcje buduje się z symboli alfabetu wejściowego konkretnej maszyny cyfrowej; przyjmiemy tu za [1] następujący alfabet:

ABCDEFGHIJKLMNOPQRSTUVWXYZ

0123456789+—/*.,"=()$

Występujący powyżej między cyfrą 9 i znakiem + odstęp jest symbolem odstępu, zwanym inaczej spacją.

Każda instrukcja określa pewną *regułę* języka (ang. *rule)* odnoszącą się do wybranego napisu. Ogólna postać instrukcji jest następująca:

[<*etykieta*>] <*reguła*> [<*określenie skoku*>]

Nie wszystkie części instrukcji muszą zawsze w niej występować, oznaczyliśmy to umieszczając w nawiasach kwadratowych te części, które można w poszczególnych instrukcjach opuszczać.

Etykieta jest to dowolny ciąg symboli, oprócz symbolu odstępu, rozpoczynający się od litery lub cyfry, np. A31, 2.5, WERSJA125, DRUGIPRZYPADEK. Jeżeli w instrukcji występuje na początku etykieta, to wymaga się, aby jej pierwszy symbol znajdował się w pierwszej kolumnie formularza kodowania lub karty dziurkowanej zawierającej instrukcję. Jeżeli instrukcja nie jest opatrzona etykietą, to w pierwszej kolumnie musi występować symbol odstępu.

Reguła ma następującą postać ogólną:

<*proste określenie napisu*> [<*wzorzec*>] [=[[<*wyrażenie zastępcze*>]]

Nawiasy kwadratowe mają takie samo znaczenie jak powyżej. Tak więc każda instrukcja musi zawierać proste określenie napisu, którego dotyczy operacja zdefiniowana za pomocą pozostałych części reguły, wybieranych dowolnie w zależności od życzenia programisty.

Proste określenie napisu jest to nazwa napisu albo sam napis *explicite*, umieszczany między apostrofami. *Nazwa napisu* jest to dowolny ciąg liter, cyfr, kropek, np. TEKST, ALFA, 11A,...5, NOWYTEKST, C.DALSZY itp. Jeżeli w programie występuje napis, który reprezentuje samego siebie, musi odróżniać się od pozostałych symboli języka, które z punktu widzenia translatora są też napisami. Dlatego przyjęto, że początek i koniec napisu umieszczanego w programie bezpośrednio (w przeciwieństwie do określanego nazwą) oznaczają apostrofy, które nie należą do napisu, ale też nie mogą być wewnątrz napisu.

Zatem *napis* reprezentujący w programie samego siebie może składać się ze wszystkich symboli alfabetu oprócz apostrofu. Można jednakże umieścić symbol apostrofu w napisie pamiętanym w maszynie, a także go wydrukować; służy do tego celu specjalna nazwa napisu QUOTE, której użycie w programie jest równoważne definicji apostrofu.

Wzorzec jest to złożone określenie napisu. *Złożone określenie napisu* jest to proste określenie napisu lub konkatenacja złożonych określeń napisów. Przykłady:

"KOT"

N.TEKST " " "KOT"

"+" " " "3"

Konkatenację oznacza się w programie w SNOBOL-u symbolem odstępu. Zgodnie z ogólną zasadą, że konkatenacja napisów jest napisem, konkatenacja zarówno prostych, jak i złożonych określeń napisów jest ostatecznie prostym określeniem napisu.

W dalszej treści artykułu będziemy w definicjach pisać *określenie napisu* wszędzie tam, gdzie jest obojętne, czy chodzi o proste, czy też złożone określenie napisu, ponieważ w bardzo niewielu przypadkach nie można stosować konkatenacji określeń napisów. Wracając do powyższych przykładów, drugi z nich byłby równoważny określeniu napisu "CZARNY KOT" wówczas, gdyby N.TEKST było nazwą napisu "CZARNY"; trzeci przykład jest równoważny określeniu napisu "+ 3".

Wyrażenie zastępcze jest to złożone określenie napisu. Poszczególne części instrukcji są oddzielone symbolami odstępu (co najmniej jednym), które nie odgrywają w tym przypadku roli operatorów konkatenacji, lecz są separatorami.

Oto przykład pełnej instrukcji bez określenia skoku:

E.1 TEKST N.TEKST " " "KOT" = „RUDY PIES I ." TEKST

Jeżeli np. TEKST jest nazwą napisu „CZARNY KOT", N.TEKST nazwą napisu "CZARNY", to wzorzec w tej instrukcji jest określeniem napisu "CZARNY

KOT", a wyrażenie zastępcze określeniem napisu "RUDY PIES I CZARNY KOT".

W następnych punktach omówimy bardziej szczegółowo reguły języka SNOBOL oraz zdefiniujemy określenie skoku.

Może się zdarzyć, że jakaś instrukcja nie mieści się w jednym wierszu formularza (na jednej karcie), wówczas jej dalszy ciąg wolno napisać w wierszu następnym (następnej karcie), umieszczając w pierwszej kolumnie następnego wiersza (karty) symbol "." jako symbol przeniesienia. Nie wolno jednak przy przenoszeniu instrukcji rozbijać nazw napisów ani napisów; jeżeli jakiś napis nie mieści się w jednym wierszu, trzeba go zastąpić konkatenacją napisów.

Między instrukcjami można umieszczać wiersze (karty) *komentarzy*, rozpoczynając je od symbolu "*" umieszczonego w pierwszej kolumnie.

Tak więc jest wyraźnie zastrzeżone, że pierwszym symbolem w wierszu musi być: litera lub cyfra, będące pierwszym symbolem etykiety, symbol odstępu wówczas, gdy instrukcja nie jest opatrzona etykietą, "." wówczas, gdy wiersz zawiera dalszy ciąg instrukcji oraz "*" w przypadku wiersza komentarza.

Koniec programu należy oznaczać etykietą END; po niej może już tylko występować w tym samym wierszu etykieta tej instrukcji, od której ma rozpocząć się wykonywanie programu po jego przetłumaczeniu. Jeżeli brak etykiety po etykiecie END, to wykonywanie programu rozpocznie się od jego pierwszej instrukcji.

Tworzenie napisów — reguła zastępowania

Napisy definiuje się w programie, korzystając *z reguły zastępowania*, mającej następującą postać ogólną:

<nazwa napisu> = *<złożone określenie napisu>*

Reguła zastępowania pozwala określić postać napisu i nadać mu nazwę, za pomocą której można następnie w programie odwoływać się do niego. Oto przykłady zastosowania tej reguły:

NAPIS1 = "ALA I AS ALI"

NAPIS2 = NAPIS1

Druga z tych instrukcji spowoduje utworzenie nowego napisu "ALA I AS ALI" oraz nazwanie go nazwą NAPIS2, przy czym pierwszy napis pozostanie nienaruszony. Zatem użycie po prawej stronie symbolu "=" nazwy napisu uprzednio zdefiniowanego powoduje skopiowanie go jako wartości nowej nazwy. Podobnie, jeżeli umieścimy w programie poniższe trzy instrukcje:

ALFA = "ABCD"

BETA = "BCDE"

GAMMA = ALFA BETA

to trzecia z nich spowoduje, że powstanie nowy napis "ABCDBCDE" o nazwie GAMMA, a wartości napisów ALFA i BETA pozostaną nie zmienione. Określając napis

LICZBA = "1"

i następnie

LICZBA = LICZBA LICZBA "0"

spowodujemy utworzenie napisu "110" o nazwie LICZBA.

Z kolei następna instrukcja

LICZBA = LICZBA LICZBA "0"

spowoduje, że LICZBA będzie teraz nazwą napisu 1101100". Brak określenia napisu po prawej stronie symbolu "=" powoduje nadanie nazwy napisowi pustemu.

Oprócz definiowania napisów w programie można je też czytać do pamięci maszyny cyfrowej za pomocą jakiegoś urządzenia wejściowego (na ogół z taśmy magnetycznej), a w programie nadawać im żądane nazwy. Do tego celu stosuje się również regułę zastępowania, w której po lewej stronie symbolu "=" umieszcza się tę nazwę, jaką chcemy nadać napisowi, a po prawej stronie symbolu "=" standardową nazwę napisu SYSPIT. Wówczas z taśmy wejściowej zostanie odczytany i zapamiętany zapis 80-symbolowy. Podobnie, użycie standardowej nazwy SYSPOT po lewej stronie symbolu "=" w regule zastępowania powoduje wypisanie na urządzeniu wyjściowym (a najczęściej na drukarce) napisu, którego określenie występuje po prawej stronie symbolu "=" w tej regule. Na przykład instrukcja

SYSPOT = "W. SHAKESPEARE — " QUOTE "ROMEO I JULIA" QUOTE

spowoduje umieszczenie na arkuszu wyjściowym napisu

W. SHAKESPEARE — "ROMEO I JULIA"

Przekształcanie napisów — reguły uzgadniania i zastępowania

Reguła uzgadniania pozwala zbadać budowę napisu, sprawdzić, z jakich składa się on symboli i wyróżnić w nim pewne podnapisy. Reguła ta ma następującą postać ogólną:

<proste określenie napisu> *<wzorzec>*

Uzgadnianie napisu z wzorcem polega na sprawdzaniu, czy w badanym napisie znajduje się ciąg symboli identyczny z napisem określonym przez wzorzec. Uzgadnianie zaczyna się od pierwszego z lewej symbolu badanego napisu, a kończy się *sukcesem* z chwilą odnalezienia ciągu symboli identycznego z napisem, który określono wzorcem lub *niepowodzeniem* wówczas, gdy dojdzie się do końca badanego napisu nie znajdując napisu poszukiwanego. Przypuśćmy dla przykładu, że zastosowaliśmy następującą regułę zastępowania:

TEKST = "ZIEMIA NIEBO MORZE LAS. CHMURY"

Wówczas reguła uzgadniania

TEKST "NIEBO"

spowoduje sprawdzenie, czy w napisie o nazwie TEKST znajduje się podnapis "NIEBO"; w tym przypadku uzgadnianie zakończy się sukcesem. Jeżeli oprócz wymienionej powyżej reguły definiującej napis o nazwie TEKST użyjemy jeszcze reguły zastępowania

WODA = "MORZE"

to reguła uzgadniania

TEKST WODA

spowoduje poszukiwanie w napisie o nazwie TEKST napisu "MORZE", również zakończone sukcesem. Natomiast wykonanie reguły

TEKST ZIEMIA

zakończy się niepowodzeniem, ponieważ nie zdefiniowaliśmy napisu o nazwie ZIEMIA, lecz umieściliśmy napis "ZIEMIA" wśród składników napisu o nazwie

TEKST.

Zauważmy jeszcze, że wykonanie reguły

TEKST "MURY"

zakończy się sukcesem, natomiast próba wykonania reguły

TEKST " MURY"

spotka się z niepowodzeniem.

Reguła uzgadniania nie powoduje żadnych zmian w strukturze badanego napisu. Służy do tego celu znana nam już reguła zastępowania, którą można połączyć z regułą uzgadniania w jednej instrukcji — dopisując po odstępie za ostatnim symbolem wzorca symbol "=", a po odstępie za nim wyrażenie zastępcze. Zastosowanie tych dwóch reguł w jednej instrukcji oznacza żądanie zastąpienia w badanym napisie ciągu symboli identycznego z napisem określonym wzorcem przez napis określony w wyrażeniu zastępczym.

Na przykład, jeżeli w zdefiniowanym powyżej napisie o nazwie TEKST chcemy zastąpić napis "NIEBO" napisem "JEZIORO", to wystarczy użyć instrukcji

```
TEKST "NIEBO" = "JEZIORO"
```

Zauważmy, że po tym zastąpieniu nowy napis o nazwie TEKST jest dłuższy od poprzedniego, lecz wszystkie odstępy między wyrazami pozostały zachowane:

"ZIEMIA JEZIORO MORZE LAS CHMURY"

Jeżeli uzgadnianie zakończy się niepowodzeniem, to reguła zastępowania się nie wykona. Tak więc w przypadku użycia instrukcji

```
TEKST "DOLINA" = "RZEKA"
```

napis o nazwie TEKST pozostaje nie zmieniony.
W przypadku użycia w instrukcji reguły zastępowania, proste określenie napisu w regule uzgadniania musi być nazwą napisu, natomiast może ono być również napisem wówczas, gdy w instrukcji brak symbolu "=", np.

"ZIEMIA NIEBO MORZE LAS" WODA

Brak w instrukcji wyrażenia zastępczego po symbolu "=" oznacza żądanie zastąpienia napisem pustym napisu identycznego z określonym przez wzorzec.

Określenie skoku

Instrukcje programu w SNOBOL-u wykonują się zazwyczaj kolejno, począwszy od instrukcji pierwszej w programie lub od opatrzonej etykietą wymienioną po etykiecie END. Można jednakże zmieniać kolejność wykonywania instrukcji za pomocą określenia skoku umieszczanego na końcu instrukcji i poprzedzanego symbolem "/", wskazującym początek określenia skoku.

Reguła uzgadniania może posłużyć do warunkowej zmiany kolejności wykonywania instrukcji w zależności od tego, czy uzgadnianie zakończyło się sukcesem, czy też niepowodzeniem. *Określenie skoku* może przyjmować jedną z opisanych niżej czterech postaci:

(1) S(<*etykieta*>)

gdzie etykieta jest etykietą instrukcji, do której ma nastąpić skok wykonywany w przypadku zakończenia operacji uzgadniania sukcesem; w przypadku przeciwnym wykona się następna kolejna instrukcja.

(2) F(<*etykieta*>)

gdzie *etykieta* jest etykietą instrukcji, do której ma nastąpić skok wykonywany w przypadku zakończenia operacji uzgadniania niepowodzeniem; w przypadku przeciwnym wykona się następna kolejna instrukcja.

(3) F(<*etykieta1*>)S(<*etykieta2*>)
lub
S(<*etykieta1*>)F(<*etykieta2*>)

Jest to złożenie dwóch określeń skoków w jednej instrukcji. W przypadku instrukcji zawierającej oba określenia skoków nie wykona się nigdy instrukcja po niej następna (o ile nie jest opatrzona etykietą wymienioną w określeniu skoku).

(4) (<*etykieta*>)

Jest to określenie skoku bezwarunkowego, w istocie równoważne postaci (3) określenia skoku warunkowego, w której obie etykiety są identyczne.

W celu ilustracji zastosowania skoków warunkowych do tworzenia *pętli* (powtarzanego wykonywania instrukcji) w programie podamy teraz przykład bardzo prostego programu służącego do zamiany wszystkich liter "A" w jakimś napisie przez litery "B" oraz wszystkich liter "B" (pierwotnie występujących w tym napisie) przez litery "A". Przyjmiemy, że napis będzie czytany ze standardowego urządzenia wejściowego i dwukrotnie przepisywany na standardowym urządzeniu wyjściowym — przed oraz po przetworzeniu. Założymy również, że w czytanym napisie nie występuje symbol "*".

```
   X = SYSPIT
   SYSPOT = "TEKST DO PRZETWORZENIA " X
1  X "A" = "*" /S(1)
2  X "B" = "A" /S(2)
3  X "*" = "B" /S(3)
   SYSPOT = "TEKST PRZETWORZONY " X
END
```

Operacje arytmetyczne

W SNOBOL-u nie istnieją wprawdzie dane typu arytmetycznego, jednakże niektóre napisy mogą mieć wartość liczb całkowitych i można na nich wykonywać oprócz takich operacji, jak na wszystkich napisach, operacje arytmetyczne. Własność tę mają wszystkie napisy składające się wyłącznie z cyfr poprzedzonych co najwyżej jednym znakiem "+" lub "—", a także napis pusty, który ma wartość zero.

Można wykonywać pięć operacji arytmetycznych: dodawanie, odejmowanie, mnożenie, dzielenie całkowite oraz potęgowanie, oznaczając je odpowiednio symbolami +, —, *, /i **. Wszystkie te operacje są dwuargumentowe. O tym, czy dane dwa napisy mają być traktowane (na chwilę wykonywania operacji) jak liczby, decyduje umieszczenie między ich określeniami symbolu odpowiedniej operacji arytmetycznej nie otoczonego apostrofami, ale zawsze oddzielonego symbolami odstępów. Wynik operacji arytmetycznej jest zawsze napisem, który również może mieć wartość liczby całkowitej. Zatem, jeżeli użyjemy np. następujących instrukcji:

```
2 = "2"
Y = "125".
WYNIK = Y + 2
```

to WYNIK będzie nazwą napisu "127". Jeżeli natomiast użyjemy instrukcji

```
WYNIK = Y "+" 2
```

w której WYNIK jest nazwą nadaną konkatenacji trzech napisów, to otrzymamy napis "125+2", który nie może mieć wartości liczbowej.

Jeżeli chcemy utworzyć napis będący wynikiem wykonania kilku operacji arytmetycznych, to trzeba wówczas pary argumentów poszczególnych operacji otaczać nawiasami, np.

```
"5" + ("3"* "2")
"20" * ((K + "3")/((N + "2") + "3"))
```

W przypadku, gdy jakieś określenie napisu jest konkatenacją napisów, między którymi występują symbole operacji arytmetycznych, najpierw są wykonywane operacje arytmetyczne, a następnie tworzy się napis będący konkatenacją napisów powstałych w wyniku tych operacji.

A więc określenie napisu

```
"3" * "2" "10" / "2"
```

jest równoważne napisowi "65". Jeżeli jednak umieścimy w tym określeniu napisu dwa nawiasy

"3" *("2" "10" / "2")

to operacja mnożenia nie może być wykonana, zanim nie będzie utworzony napis będący jej drugim argumentem i wówczas takie określenie napisu jest równoważne napisowi "75".

Możliwość wykonywania operacji arytmetycznych w SNOBOL-u jest jedną z tych własności, których brak zarzucano językowi COMIT, okazało się bowiem, że podczas wykonywania rozmaitych przekształceń napisów często trzeba też coś policzyć, np. liczbę symboli usuwanych z przetwarzanego tekstu. Zilustrujemy to prostym programem:

```
        TEKST = SYSPIT
SZUKAJ  TEKST "," = /F(KONIEC)
        LICZNIK = LICZNIK + "1" /(SZUKAJ)
KONIEC  SYSPOT = " , WYSTAPIL " LICZNIK
        RAZY"
END
```

Po wykonaniu tego programu, jeżeli w napisie było np. 15 przecinków, na standardowym urządzeniu wyjściowym będzie napisany tekst:

, WYSTAPIL 15 RAZY

Zmienne napisowe

Podczas rozmaitych przekształceń napisów często zdarza się, że trzeba wyodrębnić jakąś część napisu, o której wiadomo np. tylko tyle, że znajduje się między określonymi symbolami. W celu nazywania takich nie znanych jeszcze napisów wprowadzono w SNOBOL-u możliwość definiowania we wzorcach tzw. zmiennych napisowych.

Zmienne napisowe są to nazwy napisów, które za pierwszym wystąpieniem we wzorcu muszą być umieszczone między dwoma symbolami "*", a za każdym następnym wystąpieniem w tej samej instrukcji, a nawet w tym samym wzorcu tracą otaczające je gwiazdki. W dalszej treści programu można je już traktować tak samo, jak wszystkie nazwy napisów. Przypuśćmy, że chcemy z jakiegoś napisu nazywanego TEKST, np. ze zdania w języku polskim, wybrać wyraz (tj. część napisu otoczoną odstępami) znajdujący się na prawo od zadanego, określonego nazwą A. Można to wykonać za pomocą jednej instrukcji:

TEKST A " " *P* " "

Po wykonaniu tej instrukcji P będzie nazwą napisu identycznego z pierwszym wyrazem znajdującym się na prawo od pierwszego wystąpienia wyrazu zadanego w tekście. W celu wyodrębnienia wyrazu znajdującego się najbliżej na lewo od zadanego trzeba już użyć dwóch instrukcji:

```
TEKST *L* " " A
POWT. L*L.STRONA* " " *P.STRONA* = P.STRONA
/S(POWT.)
```

Instrukcja opatrzona etykietą POWT. będzie powtarzana dopóty, dopóki z napisu o nazwei L nie będzie można wyodrębnić dwóch napisów oddzielonych odstępem, a wówczas napis ten (jak również napis o nazwie P.STRONA) będzie poszukiwanym wyrazem. Jak widać, w wyniku użycia zmiennej napisowej tworzy się nowy napis oraz nazwa tego napisu. Czasami taki napis nie jest już potrzebny dłużej po wyodrębnieniu go z większego napisu, np. wówczas, gdy usuwa się go zastępując innym. Nie warto więc określać nowej nazwy, skoro nie będzie więcej używana. W takim przypadku można definiować tzw. *zmienną napisową nie nazwaną*, którą oznacza się za pomocą dwóch symboli gwiazdki umieszczonych bezpośrednio obok siebie. Na przykład, jeżeli wiemy, że jakiś napis o nazwie NAPIS składa się ze znanych napisów o nazwach X i Y rozdzielonych jakimś nieznanym ciągiem symboli, to instrukcja

NAPIS X ** Y = X Y

spowoduje zastąpienie tego napisu konkatenacją wyodrębnionych zeń napisów o nazwach X i Y, czyli usunie z niego napis zawarty między tamtymi dwoma.

Za pomocą zmiennych napisowych zarówno nazywanych, jak i nie nazywanych można tworzyć napisy o określonej z góry długości. W tym celu umieszcza się w zmiennej napisowej za nazwą napisu, ale przed drugą gwiazdką określenie długości napisu poprzedzone symbolem "/".

Określenie długości napisu jest to proste określenie napisu przejmującego wartość liczby całkowitej. Np. jeżeli chcemy wyodrębnić pierwszy symbol w napisie o nazwie TEKST, to możemy zastosować instrukcję:

TEKST *X/"1"*

X będzie wówczas nazwą napisu składającego się z pierwszego symbolu napisu o nazwie TEKST. Inne przykłady zastosowania zmiennych napisowych o określonej długości będą przytoczone poniżej.

Określenie pośrednie napisu

Oprócz opisanych dotychczas możliwości tworzenia nowych nazw napisów, w SNOBOL-u istnieje jeszcze jeden sposób, zwany *określeniem pośrednim*. Jeżeli w pewnym programie języka SNOBOL napis o nazwie, powiedzmy A, jest sam nazwą pewnego innego napisu, to do tego innego napisu możemy się odwołać przez symbol $A stanowiący tzw. określenie pośrednie napisu. Jeżeli więc w pewnym programie występują reguły:

A = "B"

B = "C"

to $A jest nazwą napisu "C", podczas gdy A, jak wiemy, jest nazwą napisu "B". Znak $ nazywamy tutaj *symbolem określenia pośredniego*. Można zauważyć analogię pośredniego określania napisów w SNOBOL-u z adresowaniem pośrednim, stosowanym powszechnie w językach symbolicznych.

Symbolem określenia pośredniego można poprzedzać określenie napisu z tym zastrzeżeniem, że złożone określenie napisu trzeba otoczyć nawiasami i symbol $ umieszczać bezpośrednio przed nawiasem otwierającym; można zatem napisać $("S" K), $("123" X "ALFA"), a także $($($A)).

Określenie pośrednie napisu może występować w dowolnym miejscu instrukcji, z wyjątkiem etykiety umieszczonej na początku instrukcji (nie jest ona bowiem określeniem napisu), może nawet występować w określeniu skoku, pozwalając np. — tak jak w poniższym programie — wybierać skoki do rozmaitych instrukcji.

```
POWT SYSPIT* KOD/"1" *          * C.D. * /($KOD)
0       POCZ. = POCZ. "." C.D.          /POWT
1       TEKST = TEKST C.D.              /(POWT)
2       SYSPOT = "KONIEC"
END
```

Pierwsza instrukcja tego programu powoduje rozbicie przeczytanego napisu na jednosylabowy napis o nazwie KOD oraz napis o nazwie C.D., zawierający pozostałe symbole. Następnie w zależności od tego, czy KOD będzie nazwą napisu "0", "1", czy też "2", nastąpi skok do instrukcji opatrzonej etykietą 0, 1 lub 2. Jeżeli napis ten nie będzie zawierał symbolu identycznego z żadną spośród etykiet programu, to wykonanie programu zakończy się z powodu źle zdefiniowanego określenia skoku.

Określenie pośrednie może też występować jako zmienna napisowa. Jako przykład może tutaj posłu-

żyć program, który ma spowodować przeczytanie 10 napisów, wybranie z każdego z nich napisów zawierających po 20 symboli, nazwanie tych napisów odpowiednio nazwami: A1, A2, ..., A10 oraz wydrukowanie każdego z utworzonych napisów poprzedzonego jego nazwą:

```
      K = "1"
KOLEJNA KARTA = "A" K
      SYSPIT * $KARTA/"20" *
      SYSPOT = KARTA $KARTA
      K "10"              /S(END)
      K = K + "1"         /(KOLEJNA)
END
```

Funkcje standardowe

W SNOBOL-u istnieje około trzydziestu funkcji standardowych, które można traktować jako z góry określone operacje na napisach będących *argumentami* funkcji, dające w wyniku inne napisy, nazywane *wartościami* funkcji. Do funkcji można się odwoływać za pomocą ich nazw, umieszczanych w dowolnych częściach instrukcji, z wyjątkiem tych miejsc, w których może występować wyłącznie nazwa napisu oraz miejsca etykiety na początku instrukcji, a więc wszędzie tam, gdzie może występować napis explicite. Odwołując się do funkcji podaje się za jej nazwą i nawiasem otwierającym (bez odstępów) argumenty przedzielane przecinkami, po ostatnim zaś argumencie umieszcza się nawias zamykający. Argumentami mogą być złożone określenia napisów, w których mogą występować również nazwy funkcji. Po odwołaniu do funkcji otrzymuje się, oprócz wyniku, tj. napisu, informację o tym, czy wykonanie jej zakończyło się sukcesem, czy też niepowodzeniem. Dla każdej bowiem funkcji określono jakiś warunek, spełnienie którego jest traktowane jako sukces, niespełnienie — jako niepowodzenie. Można dzięki temu korzystać z funkcji w celu określenia warunkowego wykonania pewnych instrukcji oraz wybierania skoków w programie.

W tym artykule nie będziemy omawiać wszystkich funkcji standardowych — zajęłoby to zbyt wiele miejsca. Wspomnimy tylko o niektórych, charakteryzując równocześnie pewne grupy funkcji.

Funkcje spełniające rolę operatorów relacji. Funkcje te służą do sprawdzania, czy zachodzą pewne relacje między napisami określanymi za pomocą argumentów. Wartością jest zawsze napis pusty. Należą do nich m.in.:

EQUALS(X, Y) — wykonanie jej kończy się sukcesem, jeżeli napisy określone argumentami są identyczne

.EQ(X, Y) — wykonanie jej kończy się sukcesem, jeżeli napisy określone argumentami mają takie same wartości liczbowe

.LE(X, Y) — wykonanie jej kończy się sukcesem, jeżeli napis określony pierwszym argumentem ma wartość liczbową mniejszą od napisu określonego drugim argumentem.

Jeżeli w wywołaniu jeden argument pominięto, to przyjmuje się, że jest nim napis pusty. Funkcje dotyczące porównań wartości liczbowych napisów powodują niepowodzenie również w przypadku, gdy któryś argument nie ma wartości liczby całkowitej.

Funkcje o wartościach liczb całkowitych. Są to m.in.:

SIZE(X) — jej wartością jest napis o wartości równej długości napisu określonego argumentem

REMDR(X, Y) — jej wartością jest napis będący liczbą równą reszcie z dzielenia liczb — wartości napisów określonych argumentami.

Funkcje wejścia/wyjścia. Oprócz standardowych urządzeń wejścia/wyjścia można jeszcze korzystać w programie z urządzeń dodatkowych, określanych za pomocą drugiego argumentu funkcji READ(X, Y), PRINT(X, Y) czy też PUNCH(X, Y). Pierwszym argumentem jest nazwa napisu, którą można się posługiwać w programie w sposób analogiczny do nazw SYSPIT i SYSPOT. Jednocześnie użycie tego typu funkcji spełnia rolę deklaracji wiążącej określoną jednostkę wejścia/wyjścia z dowolnie wybraną przez programistę nazwą napisu.

Funkcje ułatwiające uruchamianie programu. Należy do nich funkcja TRACE(X), gdzie X oznacza ciąg nazw funkcji definiowanych w programie (patrz punkt następny) powodująca drukowanie nazw funkcji wymienionych w X oraz wartości ich argumentów, a także numerów poziomów wywołania tych funkcji w przypadku rekursji. Podczas wykonywania programu wszystkie te informacje są drukowane za każdym odwołaniem do funkcji wymienionej w X. Funkcja STOPTR(Y), gdzie Y oznacza ciąg nazw funkcji definiowanych w programie, powoduje zaprzestanie wykonywania funkcji TRACE dla tych jej argumentów, które wymieniono w Y. Funkcja STRACE(X), gdzie X jest nazwą napisu, powoduje drukowanie napisu określonego argumentem, wykonywane w przypadku każdej zmiany dokonanej w jego budowie podczas wykonywania programu.

Funkcje definiowane w programie

Oprócz funkcji standardowych, realizowanych przez programy napisane przez twórców translatora SNOBOL-u i zawsze dostępnych dla użytkownika tego języka, mogą w każdym programie występować funkcje definiowane przez programistę.

Sposób korzystania z tych funkcji jest identyczny, jak w przypadku funkcji standardowych, trzeba tylko spełnić dodatkowo dwa warunki: zadeklarować nazwę i argumenty funkcji oraz dołączyć do programu jej kod — ciąg instrukcji w SNOBOL-u lub w języku maszyny.

Deklarację funkcji określa się za pomocą standardowej funkcji DEFINE. Postać ogólna deklaracji funkcji jest następująca:

DEFINE("<*nazwa funkcji*>(<*argumenty*>)","<*etykieta*>"[,"*nazwy*"])

Nazwą funkcji może być dowolna nazwa napisu, *argumenty* oznaczają ciąg argumentów oddzielanych przecinkami, *etykieta* oznacza etykietę, od której rozpoczyna się kod funkcji, *nazwy* oznaczają wymieniane ewentualnie (odpowiednia część w postaci ogólnej deklaracji jest ujęta w nawiasy kwadratowe) nazwy napisów lokalnych dla kodu funkcji.

Argumenty umieszczane w deklaracji funkcji są *formalne*, tzn., że występują w kodzie funkcji, lecz przy odwołaniu do funkcji ich wartościami stają się argumenty *aktualne* — kopie napisów określonych w odwołaniu; wykonanie funkcji nie zmienia wartości argumentów aktualnych.

Wartością funkcji jest napis o nazwie identycznej z nazwą funkcji, musi zatem w kodzie funkcji występować instrukcja przyporządkowująca tej nazwie jakąś wartość. Zastrzega się, że nie wolno do żadnej instrukcji kodu skakać z zewnątrz kodu ani też korzystać z innego wyjścia z kodu powrotu (do programu głównego) niż za pomocą standardowych etykiet: RETURN (w przypadku wyjścia przy sukcesie) i FRETURN (przy niepowodzeniu), z których przynajmniej jedna musi występować w określeniu skoku jakiejś instrukcji kodu. Deklaracja funkcji musi być umieszczona w programie wcześniej niż odwołanie do jej nazwy, natomiast kod może znajdować się przy końcu programu.

W SNOBOL-u można definiować funkcje rekursywnie, można też w sposób rekursywny odwoływać się do wszystkich funkcji — standardowych i definio-

wanych. Poniżej przytoczymy przykład deklaracji, kodu i odwołania dla funkci n!, nazywanej tutaj FACT(N).

Deklaracja:

```
DEFINE("FACT(N)", "ZERO")
```

Kod:

```
ZERO    FACT = .EQ(N, "0") "1"          /S(RETURN)
        FACT = FACT(N — "1") * N        /(RETURN)
```

Odwołanie w programie:

```
SYSPOT = "25! = " FACT("25")
```


BIBLIOGRAFIA

[1] E. J. Desautels and D. K. Smith — „An introduction to the string processing language SNOBOL; Programming Systems and Languages", ed. by Saul Rosen (1967), pp. 419—454

[2] D. J. Farber, R. E. Griswold and I. P. Polonsky — „SNOBOL, a string manipulation language", Journal of the ACM (Jan. 1964), V. 11, nr 1

[3] H. Katzan, Jr — „Advanced Programming, Programming and Operating Systems', (1970), pp. 36—48

[4] J. Koncewicz — „Języki do przetwarzania struktur listowych", „Maszyny Matematyczne" nr 5/1970

[5] J. E. Sammet — „Programming Languages" — History and Fundamentals (1969), pp. 436—448

J. NIEVERGELT i J. C. FARRAR
Uniwersytet Illinois,
Urbana, USA

Co maszyny mogą, a czego nie mogą robić?

Część 1

1. Wprowadzenie

Pytanie „Co maszyny mogą, a czego nie mogą robić?" zadano po raz pierwszy wcale nie w związku z komputerami. Ludzi zawsze zastanawiało, jak wiele mogą zdziałać maszyny typowe dla ich czasów. Największe zainteresowanie wzbudzały maszyny do pewnego stopnia naśladujące działanie człowieka. Zawsze byli tacy, którzy wymarzyli, następnie zaprojektowali, a czasem nawet zrealizowali budowę urządzenia tego typu, co napędzany parą wojownik, czy poruszany mechanizmem zegarowym mechaniczny skryba.

Olbrzymi wzrost szybkości obliczeń i budowanie coraz bardziej skomplikowanych maszyn w ostatnim dwudziestoleciu oraz coraz większe umiejętności tworzenia algorytmów kierujących pracą maszyny zgodnie z naszym życzeniem sprawiły, że znaczenie tego zagadnienia jest znacznie większe obecnie niż było kiedykolwiek dotychczas. Stale wzrastająca rola komputerów i sterowanych przez nie maszyn wymaga, aby społeczeństwo dowiedziało się o możliwościach i granicach możliwości działania maszyn.

Nie ma jednoznacznej odpowiedzi na pytanie, które zadaliśmy w tytule, chociażby tylko z tego powodu, że słowa „może zrobić" nie są jednoznaczne i można je różnie interpretować. Jednak w ciągu ostatnich czterdziestu lat sformułowano kilka bardziej ścisłych pytań związanych z pytaniem przytoczonym powyżej i częściowo na nie odpowiedziano. Poznanie tych pytań i ich zrozumienie pomoże w uniknięciu niejasności co do możliwości maszyn i rozproszy pewne rozprzestrzenione na ten temat mity.

2. Czy maszyna może myśleć?

Test Turinga

Prowokacyjne pytanie „Czy maszyna może myśleć?" jest prawdopodobnie jednym z najnowszych pytań na temat maszyn. Do momentu powstania komputerów cyfrowych maszyny przeznaczone były wyłącznie do celów czysto mechanicznych, to zaś nie stwarzało powodów do dyskusji o ich możliwościach intelektualnych.

Były jednak już dawniej przypadki, które sprawiły, że ludzie zaczęli się zastanawiać nad myśleniem maszyn. Przykładem może być zrobiony przez barona von Kemplena około roku 1800 fałszywy automat grający w szachy. Ten pokazywany przez wiele lat robot miał opinię niezwyciężonego. W jego środku siedział człowiek tak schowany, że chociaż widzom wolno było zaglądać kolejno do wszystkich miejsc wewnątrz maszyny, mógł on przechodząc z miejsca na miejsce uniknąć wykrycia.

Jest obecnie kilka programów szachowych [1] grających na poziomie wstępnych rozgrywek turniejowych, zdolnych do pokonania większości ludzi grających w szachy tylko sporadycznie. Istnieje także program gry w warcaby [2], który osiągnął poziom mistrzowski.

Większość ludzi zgodzi się, że jest to niebanalne osiągnięcie intelektualne. Bardzo często jednak dyskusja na temat możliwości wykonywania przez maszyny zadań intelektualnych zostaje zmącona nierzeczowymi emocjonalnymi argumentami.

Nad zagadnieniem maszyn myślących zastanawiał się znany logik A. M. Turing; po dziś dzień jego praca z 1950 r. najlepiej wyjaśnia to zagadnienie [3].

Turing uważał, że pytanie „Czy maszyna może myśleć?" ma charakter emocjonalny i odpowiedź na nie zależy od subiektywnej interpretacji pojęcia myślenia. Wykazał on, że dla pewnych interpretacji odpowiedź na to pytanie nie jest kwestią logicznych argumentów, ale sprawą wiary. Niektóre argumenty na temat myślących maszyn są oparte na ukrytym założeniu, że myślenie z definicji jest tym, co robią ludzie, zatem maszyny myśleć nie mogą, chyba, że będzie się dyskutowało, czy maszyny są ludźmi lub, czy ludzie są maszynami, a tego nie mamy zamiaru robić.

Turing zaproponował zastąpienie starego pytania nowym, po to, aby „myślenie" maszyny mogło być dyskutowane wyłącznie w kategoriach technicznych. Od tego czasu rozumowanie to nosi nazwę testu Turinga i przeprowadza się je w następujący sposób: Wyobraźmy sobie komputer — C i człowieka — M znajdujących się w jednym pokoju i podłączonych

do dwu dalekopisów w innym pokoju, w którym jest także osoba przeprowadzająca test — I. Zadaniem I jest określenie, który dalekopis podłączony jest do M, a który do C. Musi on to zrobić na podstawie odpowiedzi, które dostaje na pytania drukowane na jednym lub obu dalekopisach. Zatem zadanie wprowadzenia I w błąd będzie testem dla komputera: C powinien starać się odpowiedzieć na pytania I w taki sposób, aby I nie mógł odróżnić C od M [4]. Tak więc pytanie, którym Turing zastąpił stare pytanie „Czy maszyna może myśleć?", brzmi: „Czy istnieją maszyny tak zbudowane, by mogły sobie dobrze radzić w tej grze?"

Oczywiste jest, że tak sformułowane pytanie różni się od oryginalnego charakterem, cóż bowiem mamy na myśli mówiąc „dobrze sobie radzić"? Nie ma odpowiedzi „tak" lub „nie", można jedynie określić, w jakim stopniu maszyna daje sobie radę. Kryterium sukcesu może być statystyczne. Przeprowadzający próbę I ma średnio p procent szans wybrania właściwej odpowiedzi po m minutach trwania próby. Istotne jest, że wyniki testu Turinga zależą od zaprojektowania maszyny, od tego, czy potrafimy wykonać maszynę tak, aby mogła brać udział w tej grze. Jakie będą możliwości maszyn w tym zakresie za 10 lub 20 lat? Nieporozumienia na te tematy wciąż jeszcze są możliwe, jednak emocje towarzyszące oryginalnemu pytaniu zniknęły całkowicie.

W pracy z 1950 roku Turing twierdzi, że przy końcu tego stulecia będzie można budować maszyny, które będą sobie tak dobrze radziły w omawianej grze, że przeciętny I prowadzący test będzie miał najwyżej 70% szans uzyskania właściwej identyfikacji po 5 minutach gry.

Nie znamy nikogo, kto by próbował wykonywać test Turinga dla jakiejś maszyny. Dużo jednak pracy włożono w wyjaśnienie, jak należy programować komputer, żeby stosunkowo dobrze potrafił prowadzić rozmowę za pomocą drukowania słów w jakimś języku naturalnym. Programy napisane z myślą o prowadzeniu pozornie mądrej konwersacji w wąskiej dziedzinie są najbardziej interesujące. Omówimy dwa takie programy napisane ostatnio. Ocenę dokładności przewidywań Turinga pozostawiamy Czytelnikowi po obejrzeniu tych przykładów.

2.2. Programy konwersacyjne

Komputery, dla których ułożono programy konwersacyjne, odczytują zdania i odpowiadają na nie w języku naturalnym, naśladując prowadzenie pozornie prawdziwej, czasem pożytecznej rozmowy na tematy z jakiejś ograniczonej dziedziny zagadnień.

Mamy zamiar zorientować pobieżnie Czytelnika, jak działają tego rodzaju programy, aby zrozumiały stał się związek między działaniem programu a jego złożoną budową. Czytelnik będzie wówczas mógł sam ocenić, czy można napisać program, który poradzi sobie z testem Turinga; jakie są możliwości obecnie w tym zakresie, a jakie będą za dziesięć lat.

Program, który dobrze naśladuje rozmowę powinien także pomóc naśladować dwa istniejące sposoby prowadzenia przez ludzi konwersacji. Po pierwsze musi zrozumieć dane stwierdzenie lub pytanie niezależnie od jego formy, tzn. musi zrozumieć sens tego zdania. Po drugie powinien prowadzić pozornie rzeczową rozmowę na dany temat nawet wówczas, gdy nie zna faktów, które by pozwalały na wyraźne odpowiedzi lub stwierdzenia. Inaczej mówiąc powinien móc udzielać pozornie rzeczowych odpowiedzi na pytania w rodzaju „Co sądzisz o Picassie?"

Różnica pomiędzy „zrozumieniem", a „pozornie słuszną" odpowiedzią nie jest wyraźna. Każdy Czytelnik zna zapewne z własnego doświadczenia dwa rodzaje nastrojów, do których próbujemy znaleźć analogię: z jednej strony wysiłek podejmowany w celu zrozumienia istoty zagadnienia, z drugiej zaś strony nastrój, gdy dyskusja nas nie interesuje, ale aby zachować się grzecznie kiwamy głową i mówimy niezobowiązujące zdania w rodzaju „Tak, to bardzo interesujące".

Charakterystyczne dla obecnego stanu wiedzy w tej dziedzinie jest to, że istniejące programy konwersacyjne starają się kłaść nacisk tylko na jeden z tych aspektów, a nie na obydwa jednocześnie.

Wciąż jeszcze jesteśmy na etapie, gdy ludzie prowadzący badania starają się ulepszać jedną metodę, zamiast wprowadzać do tego samego programu kilka metod naraz.

Spośród kilku ostatnio napisanych programów najlepiej ilustrują te dwie krańcowości (konwersacja i niewiele zrozumienia oraz zrozumienie i niewiele konwersacji) dwa programy o nazwach ELIZA [5] i STUDENT [4].

ELIZA podtrzymuje konwersację o pozorach zrozumienia bez próby „rozumienia" dyskutowanego tematu. Słowo „zrozumienie" użyte jest tu w następującym sensie: jeżeli człowiek zachowałby się w sposób poniżej opisany powiedzielibyśmy, że niewątpliwie odpowiada on bez zrozumienia tematu.

ELIZA jest programem ogólnego przeznaczenia, który akceptuje zdania po angielsku i sam odpowiada zdaniami utworzonymi przez pewne przekształcenie zdań wejściowych. Przekształcenie to jest określone przez pewien „skrypt", a zachowanie ELIZY znacznie się zmienia zależnie od rodzaju zastosowanego skryptu. Jeden z takich skryptów o nazwie DOCTOR prowadzi dialog tak, jakby to robił psychiatra. Przytoczymy tutaj jedną z takich znakomitych konwersacji. Wypowiedzi ELIZY podane są dużymi literami.

Wszyscy mężczyźni są jednakowi
W JAKIM SENSIE
Zawsze wiercą nam o coś dziurę w brzuchu
CZY MASZ NA MYSLI JAKIS SZCZEGOLNY PRZYPADEK
Tak, mój chłopak skłonił mnie do przyjścia tutaj
TWOJ CHLOPAK SKLONIL CIE DO PRZYJSCIA TUTAJ
Twierdzi, że na ogół mam depresje
PRZYKRO MI ZE MASZ DEPRESJE
Tak. Jestem nieszczęśliwa
UWAZASZ ZE PRZYJSCIE TUTAJ POMOZE CI NIE BYC NIESZCZESLIWA
Potrzebuję jakiejś pomocy, to przynajmniej jest pewne
CO BY ZMIENILO OTRZYMANIE JAKIEJS POMOCY
Może nauczyłabym się być w zgodzie z matką
OPOWIEDZ MI O SWOJEJ RODZINIE
Matka się mną opiekuje
KTO JESZCZE Z TWOJEJ RODZINY OPIEKUJE SIE TOBA
Ojciec
OJCIEC
Jesteś do pewnego stopnia podobny do mojego ojca
JAKIE WIDZISZ PODOBIENSTWO
Jesteś niespecjalnie agresywny, ale mam wrażenie, że nie chcesz, abym to zauważyła
DLACZEGO SADZISZ ZE NIE JESTEM SPECJALNIE AGRESYWNY
Nie kłócisz się ze mną
UWAZASZ ZE DLACZEGO NIE KLOCE SIE Z TOBA
Bo się mnie boisz
CZY MYSL ZE SIE CIEBIE BOJE SPRAWIA CI PRZYJEMNOSC
Mój ojciec boi się wszystkich
CO JESZCZE PRZYCHODZI CI NA MYSL W ZWIAZKU Z OJCEM

Bzdury
CZY MA TO COS WSPOLNEGO Z FAKTEM ZE TWOJ CHLOPAK SKLONIL CIE DO PRZYJSCIA TUTAJ

Poniżej przytoczymy wyjaśnienia autora programu, J. Weizenbauma z MIT, w jaki sposób ELIZA może udzielać odpowiedzi tego typu, co wyżej przytoczone. „Główna procedura programu jest całkiem prosta. Tekst po wczytaniu jest analizowany pod kątem obecności słowa kluczowego. W przypadku znalezienia takiego słowa, zdanie zostaje przekształcone według zasady związanej z tym słowem. Jeżeli słowa kluczowego nie ma, otrzymuje się pozbawioną treści uwagę lub w pewnych warunkach przekształcenie wcześniejszego zdania. Tekst utworzony w ten sposób lub wyszukany jest następnie drukowany.

Rozważmy zdanie „Jestem ostatnio bardzo nieszczęśliwy". Załóżmy, że jakiś cudzoziemiec o ograniczonej znajomości języka, ale o bardzo dobrym słuchu usłyszał takie zdanie i zrozumiał jedynie pierwsze słowo „jestem". Chcąc wywrzeć wrażenie zainteresowania, czy nawet sympatii może on odpowiedzieć „Od jak dawna jesteś ostatnio bardzo nieszczęśliwy?" Zastosował on rodzaj szablonu w stosunku do zdania oryginalnego odcinając „jestem". Pozostaje wówczas resztka „ostatnio bardzo nieszczęśliwy". Potrzebna tu jest przystosowana do tego szablonu umiejętność „ponownego ustawienia", aby jakiekolwiek zdanie typu „Jestem BLAH" zostało przekształcone na zdanie „Od jak dawna jesteś BLAH?" niezależnie od znaczenia słowa BLAH. Nieco bardziej skomplikowanym przykładem będzie zdanie „Wydaje się, że ty mnie nie lubisz". Cudzoziemiec zrozumie w tym zdaniu „ty" i „mnie". Zastosuje więc szablon, za pomocą którego rozłoży zdanie na cztery części:

(1) wydaje się że (2) ty (3) mnie (4) nie lubisz, z których są zrozumiałe tylko druga i trzecia. Zdanie to można przekształcić w następujący sposób: „Dlaczego myślisz, że ja ciebie nie lubię", to znaczy, że zostają odrzucone słowa wchodzące do składnika pierwszego. Znane słowa „ty" i „mnie" są przekształcone na „ja" i „ciebie" i jest doczepiony na początku zwrot wzięty z pamięci (dlaczego myślisz)."

Skrypt DOCTOR składa się z około półtorej strony zasad rozbioru i składania zdań. Doprawdy jest zaskakujące, że taki stosunkowo prosty zestaw zasad może dać konwersację w rodzaju przytoczonej powyżej.

Jednak ta prostota kosztuje. W omawianym skrypcie jednym z zadań ELIZY jest zamaskowanie braku rozumienia. Gdyby nie stosować zasad rozbioru, łatwiej byłoby dać odpowiedź w rodzaju „A teraz porozmawiamy o czymś innym. Powiedz mi to i to." Tego typu zwrot, nie wymagający żadnego zrozumienia omawianego tematu, może być zawsze zastosowany w rozmowie (byle nie za często).

Drugi z omawianych programów ma nazwę STUDENT. Rozwiązuje on zadania algebraiczne tego typu, co poniżej przytoczone:

„Suma czterech liczb wynosi 9. Druga liczba jest większa o trzy od podwojonej pierwszej. Trzecia liczba jest równa sumie pierwszych dwóch. Jakie są te trzy liczby?".

„Mary ma dwa razy więcej lat niż miała Anna wówczas, gdy Mary miała tyle lat, co Anna obecnie. Jeżeli Mary ma 24 lata, ile lat ma Anna?"

Po wprowadzeniu do maszyny pierwszego zadania STUDENT drukuje:

PIERWSZA LICZBA JEST 0,5

DRUGA LICZBA JEST 4

TRZECIA LICZBA JEST 4,5

W odpowiedzi na drugie zadanie drukuje:

ANNA MA 18 LAT

Przy pisaniu STUDENT-a przyjęto, że świat składa się ze słownych opisów różnych równań algebraicznych, zwłaszcza liniowych. STUDENT po otrzymaniu opisu zaczyna identyfikować niewiadome i związki między nimi oraz tłumaczyć opis słowny na układ równań w konwencjonalnym zapisie matematycznym. W pierwszym przytoczonym zdaniu, STUDENT identyfikuje trzy niewiadome jako f (pierwsza liczba), s (druga) i t (trzecia) i tworzy następujący układ równań:

$$f + s + t = 9$$

$$s = 3 + 2f$$

$$t = f + s$$

Po osiągnięciu tego etapu reszta jest prosta, gdyż sposoby rozwiązywania układów równań liniowych są dobrze znane. My interesujemy się STUDENT-em wyłącznie pod kątem jego możliwości tłumaczenia zdań z dosyć ograniczonej angielszczyzny na równania algebraiczne wyrażone w postaci zapisu formalnego. Z konieczności uprościmy sobie omówienie szczegółów wykonywania przez STUDENT-a tłumaczenia, mamy jednak nadzieję, że zasada jego działania stanie się jasna.

STUDENT klasyfikuje słowa i zwroty w analizowanych zdaniach i dzieli je na trzy kategorie: zmienne, operatory, substytutory (variables, operators, substitutors).

Substytutory są to zwroty, które STUDENT natychmiast zastępuje innymi zwrotami. Na przykład słowo „dwukrotnie" zawsze zastępowane jest zwrotem „dwa razy". Zastępowanie to ma na celu zredukowanie liczby form lingwistycznych przez zastąpienie wielu różnorodnych konstrukcji specjalnych przez mniejszą liczbę bardziej ogólnych.

Operatory są to zwroty opisujące znane STUDENT-owi działania arytmetyczne (dodawanie, odejmowanie, mnożenie, dzielenie i podnoszenie do potęgi) lub kombinacje tych działań takie, jak „suma X i iloczynu Y i Z" oraz relację równości.

Poniżej podajemy niektóre formy lingwistyczne, których można używać w celu wyrażenia operacji arytmetycznych i relacji równości:

$x = y$	x to y; x równa się y; x i y są równe;
$x + y$	x plus y; suma x i y; o x więcej niż y;
$x - y$	x minus y; różnica między x i y;
$x * y$	x razy y; x mnożone przez y;
x/y	x dzielone przez y; x przez y;

Trzecią kategorią słów i zwrotów rozróżnianych przez STUDENT-a są zmienne. Zmienna jest to sekwencja słów reprezentująca niewiadomą danego zadania. STUDENT nie próbuje zrozumieć znaczenia zmiennych. Stara się jedynie zidentyfikować, które zwroty są zmiennymi, a które opisują tę samą niewiadomą. Jeżeli STUDENT może rozwiązać zadanie, w którym pojawia się zmienna „wiek Anny", to może także rozwiązać podobne zadanie, w którym wszystkie zwroty „wiek Anny" zostały zastąpione jakimś nie mającym sensu słowem *fgh*. Ponieważ STUDENT nie stara się zrozumieć istoty danej zmiennej, ma on bardzo ograniczone możliwości rozpoznawania różnych zwrotów opisujących tę samą zmienną. Na przykład STUDENT nie stwierdziłby, że fraza „ilość lat, które upłynęły od urodzenia Anny" opisuje tę samą niewiadomą, co „wiek Anny".

Zasadę, dzięki której STUDENT rozpoznaje równania algebraiczne występujące w postaci słów można opisać w następujący sposób: po zastąpieniu wszystkich substytutorów, STUDENT sprawdza występowanie operatorów przez porównanie zwrotów ze zdań wejściowych ze zwrotami-operatorami zmagazynowanymi w katalogu. Następnie identyfikuje sekwencje słów między operatorami jako zmienne i podejmuje decyzje, co do tego, które sekwencje opisują tę samą niewiadomą. Jak już mówiliśmy, STUDENT tylko wtedy rozpozna, że dwa zwroty opisują tę samą nie-

wiadomą, gdy zwroty te są identyczne lub różnią się tylko minimalnie.

Jeżeli zdania wejściowe były tego rodzaju, że STUDENT je rozumiał, to powstaje układ równań algebraicznych zapisanych w sposób matematyczny. Jeżeli program potrafi go rozwiązać, to czyni to i drukuje wyniki. W przeciwnym przypadku, gdy np. niektóre z równań są nieliniowe lub gdy równań jest mniej niż niewiadomych, STUDENT pyta: „Czy znasz jeszcze jakieś współzależności między zmiennymi?" Jeżeli odpowiedź będzie „nie", STUDENT odpowiada „Nie mogę rozwiązać tego zadania".

Metoda ta pozwala STUDENT-owi na osiąganie dobrych wyników. Naturalnie po zrozumieniu zasad działania STUDENT-a łatwo jest postawić problem, którego nie potrafi on rozwiązać. Jeżeli np. w jakimś zdaniu będzie zmienna „ile razy ją widziałem", STUDENT zidentyfikuje „razy" jako operator mnożenia i dojdzie do wniosku, że występują tutaj dwie zmienne „ile" i „widziałem ją".

Na podstawie zadań rozwiązanych przez STUDENT-a można jednak stwierdzić, że w wąskim zakresie algebraicznym jest on lepszy od przeciętnego licealisty. Fakt ten dodaje pewnego znaczenia słowom autora programu D. G. Bobrowa, który stwierdził: „Myślę, że daleko jeszcze do napisania programu, który będzie rozumiał cały język angielski, czy choćby tylko dużą jego część. Jednak w ramach swojego wąskiego zakresu kompetencji STUDENT wykazał, że można zbudować maszynę, która „rozumie".

Tłumaczyła
Janina Rowińska

BIBLIOGRAFIA DO CZĘŚCI I

[1] Shannon C. E. — „Programming a digital computer for playing chess", Philosophy Magazine, t. 41, 356—357, Marzec 1950.

[2] Newman J. R. (wybór prac różnych autorów) — „The World of Mathematics", t. 4, Simon and Schuster, N. Y. 1956. Greenblatt R. D., Eastlake D. E. and Crocker S. D. — „The Greenblatt chess program", AFIPS Conference Proc., t. 31 (FJCC 1967) 801—810, Thompson books, Washington, D.C., 1967.

[3] Samuel A. — „Some studies in machine learning using the game of checkers", IBM J. of Res. and Development, t. 3, nr 3, 210—229, lipiec 1954.

Samuel A. — „Some studies in machine learning using the game of checkers. II — Recent progress", IBM J. of Res. and Development, t. 11, nr 6, 601—617, listopad 1967.

[4] Turing A. M. — „Computing Machinery and Intelligence". Mind, t. 9, 433—460, 1950.

[5] Minsky M. (Red.) — „Semantic Information Processing", MIT Press, Cambridge, Mass., 1968 (w rozdz. 3 opisany jest STUDENT Bobrow D. G., „Natural Language Input for a Computer Problem-Solving System", 135—315).

[6] Weizenbaum J. — „Contextual Understanding by Computers", Comm, ACM, t. 9, nr 1, 36—45, styczeń 1966.

Weizenbaum J. — „Contextual Understanding by Computers", Comm. ACM, t. 10, nr 8, 474—480, sierpień 1967.

WIADOMOŚCI PKAPI

Plany krajowego przemysłu komputerowego na Plenum PKAPI

W dniu 25 czerwca 1971 roku odbyło się w Warszawie plenarne posiedzenie Polskiego Komitetu Automatycznego Przetwarzania Informacji Naczelnej Organizacji Technicznej, poświęcone głównie zagadnieniom rozwoju przemysłu komputerowego w Polsce.

Omawiano te problemy w dwóch pierwszych punktach porządku dziennego posiedzenia, a mianowicie:

1. Zadania przemysłu komputerowego w Polsce. Referaty wygłosili kol. kol. mgr inż. Jerzy Huk oraz mgr inż. Janusz Matejak. W dyskusji zabrało głos 15 członków Plenum. Kol. dr inż. Andrzej Targowski ustosunkował się do niektórych wypowiedzi dyskutantów. Kol. mgr inż. J. Huk odpowiedział wyczerpująco na zadane pytania i poruszone sprawy.

2. Rezultaty prac nad programem bazy sprzętowej API. Raport z prac Komisji PKAPI NOT do spraw rozwoju krajowej produkcji i dostaw sprzętu informatycznego — złożył kol. dr inż. Andrzej Janicki. Prace Komisji miały poważny wpływ na ukształtowanie się programu Zjednoczenia MERA, który zaprezentowano w punkcie 1. Plenum PKAPI przyjęło omawiany raport wraz z wnioskami, m.in. o konieczności dalszych prac nad problemem zaplecza naukowo-badawczego.

Na Plenum omówiono również sprawę przygotowania do Krajowej Konferencji Informatyki na podstawie relacji kol. dr doc. Zbigniewa Kierzkowskiego. Ustalono wstępnie termin tej Konferencji na III kwartał 1972 roku. Ponadto przyjęto raport złożony przez kol. mgr inż. Antoniego Bossowskiego, dotyczący wniosków Komisji PKAPI do spraw integracji środowiska informatyków w kraju. Wśród szeregu wniosków — poza wnioskami organizacyjnymi — zgłoszono wniosek w sprawie systematycznego przekazywania przez Zjednoczenie MERA do redakcji czasopisma „Informatyka" wiadomości o przemyśle komputerowym w Polsce — w celu publikacji.

*

Problematyka rozwoju przemysłu komputerowego w bieżącym pięcioleciu zajęła na posiedzeniu plenarnym znacznie więcej czasu niż planowano. Zebrani z dużym zainteresowaniem zaznajomili się z planami produkcyjnymi i rozwojowymi Zjednoczenia MERA, które po raz pierwszy przedstawiono na Plenum PKAPI. Plany te są wynikiem skrupulatnych opracowań i uzgodnień przemysłu z Krajowym Biurem Informatyki przy aktywnym udziale Komisji, powołanej przez przewodniczącego PKAPI prof. dr Zbigniewa Jasickiego. Obecny okres określa się jako „pięciolecie sprzętu informatyki". Ilościowo-jakościowe plany na ten okres zostały w zasadzie zapięte, a zapotrzebowanie informatyki polskiej na sprzęt powinno być w przybliżeniu pokryte w wyniku realizacji tego programu. Stwierdzili to zarówno dyrektor naczelnego Zjednoczenia MERA mgr inż. Jerzy Huk, zastępca dyrektora Generalnego Krajowego Biura Informatyki dr inż. Andrzej Targowski, jak i dr inż. Andrzej Janicki z ramienia Komisji PKAPI do spraw rozwoju bazy sprzętowej informatyki. Szereg problemów wzbudza jeszcze wątpliwości, trwają dalsze prace nad skonkretyzowaniem niektórych pozycji planu.

Przyjęty program produkcji i rozwoju techniki nakłada bardzo trudne i napięte zadania na Zjednoczenie MERA, podległe mu zakłady produkcyjne oraz na zaplecze naukowo-badawcze. O skali tych trudności świadczy fakt, że do roku 1970 produkcja sprzętu komputerowego stanowiła w Zjednoczeniu MERA tylko niewielki fragment całości jego planu produkcyjnego. I tak np. udział produkcji branży maszyn matematycznych w Zjednoczeniu MERA w 1965 roku wynosił 6%, zaś w 1975 r. wyniesie on 52%. Do 1970 roku nastąpił przyrost tej produkcji w wysokości 282%, a na 1975 rok planuje się przyrost 1200% w stosunku do roku 1970. Ta zasadnicza zmiana następuje w latach 1971—72 w wyniku koncentracji produkcji sprzętu informatyki w Zjednoczeniu MERA i dostosowania profilu produkcyjnego szeregu podległych mu zakładów wytwórczych do nowych zadań. Dotyczy to przede wszystkim takich zakładów, jak ELWRO, ZMP BŁONIE, MERAMONT, MERAMAT, ERA oraz innych.

Program rozwoju produkcji przemysłu komputerowego oparto na następujących założeniach:

- Produkcyjny ilościowo-jakościowy program przemysłu komputerowego na lata 1971—1975 powinien zapewnić realizację minimalnie określonego „Programu Rozwoju Informatyki", powinien uwzględnić poziom potrzeb krajowych (który można nazwać poziomem „wypłacalnego popytu") na systemy wykorzystujące komputery średnie, małe, minikomputery oraz automaty obrachunkowe.
- Program powinien zagwarantować wywiązanie się przemysłu z podjętych już zobowiązań w zakresie dostaw urządzeń peryferyjnych do krajów socjalistycznych, zarówno pod względem asortymentu, jak i wartości.
- Przemysł powinien wytworzyć masę towarową w postaci wybranego asortymentu sprzętu informatyki w celu rozwoju eksportu do krajów kapitalistycznych.
- Produkcję niektórych urządzeń tzw. trzeciej peryferii i wyposażenia będą prowadzić także zakłady poza przemysłem maszynowym.

ZAPLECZE NAUKOWO-BADAWCZE PRZEMYSŁU KOMPUTEROWEGO

Poważny nacisk kładzie się na dalszą rozbudowę zaplecza naukowo-badawczego, zarówno centralnego, jak i zakładowego. Tworzy się sieć placówek związanych terytorialnie z istniejącymi zakładami lub w rejonach, w których przewidywana jest rozbudowa przemysłu sprzętu informatyki. W roku 1970 utworzono oddziały Instytutu Maszyn Matematycznych w Gliwicach i Toruniu. W roku 1971 uruchomiono oddziały Zakładu Doświadczalnego IMM w Garwolinie oraz ZD w Gliwicach, rozbudowano oddział ZD ZMP BŁONIE w Zabrzu oraz zorganizowano pion maszyn matematycznych w Zakładach ERA w Warszawie.

Główne zadanie zaplecza naukowo-badawczego — to realizacja problemu węzłowego 06.3.1, tj. opracowanie i uruchomienie produkcji komputerów III generacji wraz z urządzeniami zewnętrznymi. Zadanie to jest związane z programem rozwoju środków informatyki w ramach współpracy krajów RWPG nad Jednolitym Systemem EMC.

Należy podkreślić bardzo poważne zaangażowanie obecnego potencjału Instytutu Maszyn Matematycznych w pracach nad Jednolitym Systemem EMC oraz nad przygotowaniem i uruchamianiem produkcji szeregu wyrobów, jak np.: drukarki wierszowe DW21 i DW3, pamięci taśmowe PT3, głowice GPT 2, GPT 3z i do pamięci taśmowych GL 5 do pamięci bębnowych, pamięć bębnowa PB 7, zasilacze itp. W pewnych okresach bardzo krótkie terminy przygotowania produkcji eksportowej musiały obciążyć IMM wraz z Zakładem Doświadczalnym dodatkowymi zadaniami absorbującymi potencjał IMM w 50—60%.

W przyszłości IMM przekaże większość prac o charakterze doświadczalno-konstrukcyjnym do rozbudowanego zaplecza zakładów produkcyjnych, a sam będzie koncentrował się na zagadnieniach systemowych, architektury, a przede wszystkim — na oprogramowaniu maszyn Jednolitego Systemu, na nowych perspektywicznych technologiach i związanej z tym aparaturze oraz na pracach nad oprogramowaniem użytkowym w wybranych zakresach działania.

KIERUNKI ROZWOJU PRZEMYSŁU KOMPUTEROWEGO NA LATA 1971—1975

Przewiduje się wdrożenie do produkcji następujących wyrobów:

Jednostki centralne

W klasie maszyn średniej mocy

- Maszyna cyfrowa R 30, należąca do Jednolitego Systemu EMC III generacji. Wdrożenie do produkcji nastąpi w roku 1973.
- Maszyna cyfrowa ODRA 1305 stanowiąca kontynuację linii ODRA 1304, z zastosowaniem dalszych ulepszeń strukturalnych i technologicznych właściwych maszynom III generacji. Będzie w niej wykorzystane podobnie, jak w ODRA 1304, bogate oprogramowanie użytkowe komputerów ICL 1900, a nawet znacznie bardziej rozszerzone niż w ODRA 1304. Wdrożenie do produkcji nastąpi w latach 1972—1973. W drugiej połowie pięciolecia 1971—75 maszyny te przeznaczone do przetwarzania danych, będą produkowane w zestawach w zależności od konkretnych zamówień.

Uwaga: W pierwszych latach pięciolecia 1971—1975 kontynuowana będzie produkcja komputerów ODRA 1304 i 1204 w obecnych wersjach aż do momentu wygaśnięcia popytu.

W klasie maszyn małych i mini

- W latach 1971—1972 wdroży się do produkcji mini-komputer K-202, przeznaczony głównie na eksport.
- W roku 1972 uruchomiona zostanie produkcja małej maszyny cyfrowej ODRA 1325; programowo współwymiennej z dotychczas produkowaną maszyną ODRA 1304 (1305) i przeznaczonej do automatyzacji procesów technologicznych (sterowania), do obliczeń inżynierskich oraz do lokalnego przetwarzania danych.
Będzie posiadała możliwości wykorzystywania oprogramowania maszyn serii ODRA 1204.
- W najbliższym czasie będą podjęte decyzje co do produkcji mikro-komputera MOMIK 8b, opracowanego w Instytucie Maszyn Matematycznych przede wszystkim do celów rejestracji danych i jako urządzenia kontrolnego w produkcji sprzętu informatyki.

Urządzenia zewnętrzne

Kierunki rozwoju konstrukcji i produkcji urządzeń zewnętrznych uwzględniają zarówno potrzeby krajowych systemów, opartych na maszynach produkcji krajowej, jak również uzgodnienia w sprawie wieloletnich dostaw do innych krajów socjalistycznych.

Urządzenia wejścia-wyjścia na taśmę papierową

- Kontynuowanie i wzrost ilościowy produkcji czytnika CT 1001 fotoelektrycznego o prędkości czytania 1000 znaków na sekundę
- Wdrożenie do produkcji w 1971 roku czytnika fotoelektrycznego CT 300 o prędkości czytania 200 znaków na sekundę
- Wdrożenie do produkcji w latach 1972—1973 czytnika fotoelektrycznego CT 2000 o płynnie regulowanej prędkości czytania w zakresie 200÷2000 znaków na sekundę
- Kontynuowanie i wzrost ilościowy produkcji szybkiej dziurkarki taśmy D 102 o prędkości 100 znaków na sekundę
- Wdrożenie do produkcji w 1972 roku szybkiej dziurkarki taśmy D 200 o prędkości regulowanej w zakresie 70÷200 znaków na sekundę i o podwyższonych parametrach niezawodnościowych
- Wdrożenie do produkcji w latach 1972—1973 prostych i stosunkowo wolnych (30÷50 znaków na sekundę) czytników i dziurkarek taśmy, przeznaczonych do wyposażenia mikro-komputerów i automatów obrachunkowych, układów sterowania liczbowego obrabiarek, transmisji danych itd.

Pamięci zewnętrzne

- Wdrożenie do produkcji w latach 1971—1972 nowego typu pamięci taśmowej PT 3 o prędkości przesyłania informacji 96 kzn/sek w miejsce wycofywanego z produkcji przestarzałego typu PT 2
- Wdrożenie do produkcji w 1971 roku i dalszy zasadniczy rozwój ilościowy pamięci bębnowej PB 7 o pojemności około 16 M bitów
- Kontynuacja produkcji aż do momentu zaspokojenia potrzeb pamięci bębnowej PB 204 do maszyny ODRA 1204,
- Wdrożenie do produkcji w roku 1972 pamięci bębnowej PB 304/305 do maszyn ODRA 1304 i 1305
- Uruchomienie produkcji najpóźniej w roku 1973 małej pamięci dyskowej z wymiennym pakietem dysków dla systemów mini-komputerowych
- Przewiduje się również około roku 1973 uruchomienie produkcji wolnej pamięci taśmowej, która będzie stanowiła wyposażenie mini-komputerów, a także wchodziłaby w zestaw urządzeń stacji przygotowania danych na taśmie magnetycznej
- Zasadniczy rozwój ilościowy produkcji głowic do pamięci magnetycznych.

Drukarki wierszowe

Przewiduje się tu opanowanie i rozwój produkcji całej rodziny szybkich alfanumerycznych drukarek wierszowych, w oparciu o uruchomioną już na podstawie licencji angielskiej produkcję mechanizmu ICL $666/V_3$ o prędkości 1100 wierszy/min, a mianowicie:

- Kontynuacja produkcji drukarki wierszowej DW 304/305 do maszyn cyfrowych typu ODRA 1305 i 1304
- Wdrożenie do produkcji w latach 1971—1972 drukarki wierszowej DW 204 dla ODRY 1204
- Kontynuacja i zasadniczy rozwój ilościowy produkcji drukarki DW 21 z modyfikacjami umożliwiającymi eksport
- Wdrożenie do produkcji w latach 1972—1973 i następnie zasadniczy rozwój ilościowy produkcji drukarki wierszowej typu DW 3 do maszyn Jednolitego Systemu EMC III generacji
- Przewiduje się również wdrożenie do produkcji około roku 1973 wolnej drukarki wierszowej o prędkości druku około 150—300 wierszy/min z przeznaczeniem do wyposażenia systemów mini-komputerowych, współpracy z liniami transmisji danych i stacjami przygotowania danych i ewentualnie z automatami obrachunkowymi.

Urządzenia do graficznego zobrazowania informacji

- Uruchomienie produkcji w latach 1972—1973 alfanumerycznego monitora dostosowanego do współpracy zarówno z maszynami typu ODRA, R 30, jak i mini-komputerami
- Produkowanie w oparciu o opracowanie WAT w jednym z zakładów doświadczalnych krótkich serii grafoskopów. W przypadku wzrostu zapotrzebowania na ten typ urządzenia, produkcja zostanie przeniesiona do zakładu przemysłowego.

Urządzenia transmisji danych

Przewiduje się uruchomienie produkcji multipleksorów do maszyn ODRA 1305, dostosowanych do współpracy z łączami transmisji danych, o szybkości 600/1200 bodów i zakończonych stacjami teledacyjnymi. Modemy i urządzenia protekcji będzie produkować Zjednoczenie UNITRA *). Dokonana będzie analiza możliwości wykorzystania mini-komputerów do systemów teledacyjnych.

Urządzenia do przygotowania danych

- Przewiduje się wdrożenie do produkcji w roku 1973 zestawów stacji przygotowania danych z wyjściem na stacje pamięci taśmowych i dyskowych.

Stacje te będą budowane w oparciu o jednostkę centralną mini-komputera i umożliwią jednoczesną pracę do 64 operatorek przy wprowadzaniu i weryfikacji danych.

Sprzęt pomocniczy

- Kontynuowana i rozwijana będzie produkcja elementów wyposażenia ośrodków obliczeniowych takich, jak:

instalacja sufit-podłoga, szafy do przechowywania kart perforowanych, taśm i dysków magnetycznych, wchodzące w skład tzw. sprzętu III peryferii.

SERWIS

Zjednoczenie MERA zgłosiło do resortu wniosek o powołanie odrębnego przedsiębiorstwa INFOMERA, którego zadaniem w najbliższej przyszłości będzie prowadzenie serwisu technicznego, a w dalszej kolejności — prowadzenie szkolenia i opracowywanie systemów dla użytkowników.

* * *

Dyskusja na Plenum PKAPI wykazała, że plany przedstawione przez przedstawicieli przemysłu oraz KBJ wymagają dopracowania w następujących dziedzinach:

- Utworzenie w możliwie najkrótszym terminie instytucji generalnego dostawcy i kompletacji systemów, komputerów i urządzeń peryferyjnych dla odbiorców i określenie szczegółowych zadań dla INFOMERY i zakładów produkcyjnych w tym zakresie
- Rozwiązanie problemu szkolenia kadr, który — jak do tej pory — jest tzw. „wąskim gardłem". Limity wynikające z przepisów przeznaczone na szkolenie są niewystarczające. Dostawy maszyn bardzo często nie są zgrane z okresem szkolenia. Wskazane byłoby uruchomienie centralnego ośrodka szkolącego kadry, będącego w gestii resortu wiodącego. NOT ani inne organizacje społeczne nie są w stanie przeszkolić potrzebnej liczby specjalistów, a szkolenie nie może być powierzchowne.
- Zapewnienie serwisu software dla maszyn wszystkich systemów (ODRA, RIAD)
- Zagadnienie kompatibilności software ODRA 1304, 1305 i R 30
- Produkcja automatów obrachunkowych — urządzeń niezbędnych do automatyzacji masowych prac obrachunkowych
- Zagadnienie produkcji K-202 dla potrzeb kraju
- Zagadnienie produkcji obwodów scalonych w kraju oraz pamięci dyskowych
- Klimatyzacja — problem wymagający szybkiego i radykalnego rozwiązania
- Ustalanie dokładnych terminów przydziału maszyn dla użytkowników, z czym związane są tak istotne sprawy, jak adaptacja lokalu (ośrodka), szkolenie kadry oraz uzyskanie kredytów
- Zapewnienie sprawnego uruchomiania maszyn MIŃSK i ODRA i dostarczania części zamiennych
- Określenie perspektywicznego programu działania Instytutu Maszyn Matematycznych
- Zjednoczenie MERA powinno również zwrócić większą uwagę na lepszą informację techniczną, wykorzystując przy tym prasę, radio i telewizję, ewentualnie otworzyć salony techniki na przykładzie firm zagranicznych.

Na podstawie protokołu opracowała
Dorota Prawdzic

*) W ostatnim okresie postanowiono przenieść tę produkcję do resortu łączności.

Jedno czy wiele stowarzyszeń informatyków?

Wiele dyskusji, wiele kontrowersji wywołuje ostatnio sprawa integracji środowiska informatyków. Informatycy tworzą bardzo ciekawe środowisko. Są to specjaliści o stosunkowo najniższej średniej wieku, to jedni z największych entuzjastów postępu nauki i techniki, to ludzie — co jest dzisiaj najbardziej interesujące — mogący naprawdę pomóc w trudnym dziele kierowania krajem, informatyka jest bowiem powszechnie uznana za najlepszy instrument zarządzania.

Informatycy niejednokrotnie dali już wyraz woli i możliwości wspólnego zorganizowanego działania społecznego przyczyniając się zdecydowanie do powstania nowych rodzajów sprzętu, nowych koncepcji, nowych organizacji itp. W dzisiejszym stadium rozwoju polskiej informatyki jednym z bardzo ważnych czynników wydaje się skonsolidowanie wszystkich twórczych sił wokół wykonania wielkich a można nawet nazwać historycznych, zadań w dziedzinie informatyki. Niestety, organizacja zawodowych, naukowych i społecznych środowisk informatyków pozostawia wiele do życzenia. Odbiega ona i nie pasuje do merytorycznej działalności informatyków. Zamiast integracji istnieje rozproszenie. Zamiast systemowego planowego działania istnieje spory indywidualizm. Z przeglądu dotychczas istniejących organizacji społecznych można wysnuć wniosek, że żadna nie odpowiada potrzebom współczesnego, dynamicznego działania, każda ma określone wady, żadna nie może objąć swym zasięgiem wszystkich informatyków. Największą działalnością dotychczas wykazał się Polski Komitet Automatycznego Przetwarzania Informacji (PKAPI), ale jest to tylko komitet a nie zorganizowane stowarzyszenie mające swoje koła terenowe. Pewnym uzupełnieniem tej luki jest Stowarzyszenie Elektryków Polskich, w którym działa Komisja Informatyki jednocząc te koła SEP, które działają na terenie zakładów ściśle związanych z informatyką — ale informatyka wybiega daleko poza techniczną strefę działania, która jest domeną SEP. W większym czy mniejszym ale zawsze wycinkowym stopniu zajmuje się informatyką wiele innych stowarzyszeń i organizacji — są to między innymi Komisja Automatyki i Pomiarów SEP; Stowarzyszenie Inżynierów Mechników Polskich; Towarzystwo Naukowe Organizacji i Kierowania; Stowarzyszenie Księgowych oraz szereg polskich towarzystw takich jak: Ekonomistów, Matematyków, Elektrotechniki Teoretycznej, Cybernetyki, Mechaniki Stosowanej itp. Zgłaszane są również wnioski o utworzenie Towarzystwa Naukowego Informatyków. Jedne z tych organizacji zlokalizowane są w Naczelnej Organizacji Technicznej, w której nadmierny wpływ biurokracji nie sprzyja dynamicznej działalności, jaka musi towarzyszyć nowej szybko rozwijającej się dziedzinie; drugie zaś zlokalizowane w Polskiej Akademii Nauk preferują niebezpieczną tendencję wyizolowania się z innych środowisk, a przecież nic tak nie podnosi wartości osiągnięć naukowych jak sprawdzenie ich w praktycznym zastosowaniu. Powstaje obecnie pytanie — czy wydać przewodnik po stowarzyszeniach i towarzystwach informatyków i utworzyć jeszcze kilka nowych a między nimi koniecznie jedno do koordynacji i współpracy międzystowarzyszeniowej — no bo chociażby za granicą warto by reprezentować jednolity front — czy też nie lepiej — co autor niniejszej wypowiedzi gorąco sugeruje — pokusić się o stworzenie nowej organizacji, o takim statucie, aby odpowiadał wymaganiom współczesnego życia w Polsce, o takiej strukturze, aby wszyscy — od naukowców profesorów do techników i operatorów, od ministrów do kierowników najniższych szczebli, od „wielkich systemowców" do użytkowników korzystających z informacji — znaleźli dobrą bazę do działania i mogli nawzajem konfrontować swoje poczynania w informatyce. Wzorem pewnym mogłyby tu być różne organizacje zagraniczne, kierunek wyznacza potrzeba dynamicznego działania i usunięcia dotychczasowych wad. Powstanie takiej organizacji byłoby wartościowym wkładem w dzieło odnowy PRL.

Antoni Bossowski
SEP

Sieć ZETO a dysponowanie mocą krajowych komputerów

Nie budzi dziś żadnych wątpliwości twierdzenie, że proces racjonalizacji naszej ekonomiki pociąga za sobą konieczność dysponowania odpowiednią sumą informacji, która by w prawidłowy sposób przedstawiała stan rzeczy w poszczególnych dziedzinach życia gospodarczego. Świadomość ta spowodowała zwrócenie uwagi na stosowanie ETO a praktycznym, wyciągniętym stąd wnioskiem, było nastawienie Wrocławskich Zakładów ELWRO na produkcję elektronicznych maszyn cyfrowych dla potrzeb wewnętrznych naszego kraju. Mimo, że decyzja ta odegrała przełomową rolę w zakresie stwarzania szerokich możliwości wyposażenia gospodarki narodowej w nowoczesne komputery, to jednak upłynęło zbyt mało czasu, aby stopień nasycenia gospodarki techniką obliczeniową można było uznać za zadowalający.

Aczkolwiek — jak dotąd — mało mamy urządzeń tego rodzaju, to jednak wielkość nakładów związana z nimi, licząca się na wiele setek milionów złotych tworzy potrzebę efektywnego nimi gospodarowania. Przyjąć należy, że jednym z podstawowych czynników warunkujących racjonalność wykorzystania istniejących i przyszłych zasobów komputerów jest prawidłowa organizacja dysponowania ich mocą. Dotychczas dyspozycja ta jest rozproszona w zbyt wielu ośrodkach, co pokazuje przykład Łodzi. Aktualnie Zakłady Elektronicznej Techniki Obliczeniowej w Łodzi rozporządzają jednym komputerem ODRA 1204, jednym ODRA 1304 oraz jednym ZAM 41. Oprócz tego Ośrodek Ekonomiki Normowania Pracy i Organizacji Przemysłu Lekkiego EKORNO uruchomi w najbliższym czasie jedną maszynę ODRA 1304 oraz Miejskie Przedsiębiorstwo Komunikacyjne jedną maszynę ZAM 41.

Mimo dużej sprawności nowoczesnej techniki obliczeniowej, nie można zupełnie wykluczyć awaryjności poszczególnych części składowych zestawu elektronicznych maszyn cyfrowych. Awaryjność ta może częściowo ograniczać ich sprawność lub w pewnych odcinkach czasu powodować przerwy w ich eksploatacji.

Wymagania racjonalnej eksploatacji komputerów, mimo że są one oparte na technice półprzewodnikowej wyznaczają również ich graniczny dobowy czas eksploatacji, powyżej którego praca ich nie jest wskazana. Nie można więc przyjmować, że współczynnik zmianowości pracy = 3 stanowić ma ideał, do którego należałoby zdążać przy eksploatacji komputerów.

Znacznie ważniejszym problemem wydaje się być właściwa organizacja eksploatacji z punktu widzenia poziomu techniki maszyny. Tak np. ODRA 1304 jest modelem, który ma możliwość równoległego wykonywania czterech programów.

Wszystkie te okoliczności nasuwają wniosek o wielkiej doniosłości gospodarczej zapewnienia racjonalnej eksploatacji wszystkich modeli komputerów zainstalowanych i uruchomionych dotąd na terenie kraju. Zachodzi paląca potrzeba powołania regionalnych ogniw organizacyjnych wspólnych dla wszystkich instytucji wyposażonych w komputery, które mogłyby dysponować wspólną mocą całego kompleksu komputerów na terenie danego regionu. Wydaje się, że najbardziej predysponowanym organem do sprawowania społecznego nadzoru

nad właściwym funkcjonowaniem organu dyspozycyjnego byłby Polski Komitet Automatycznego Przetwarzania Informacji — oddziały wojewódzkie NOT. Międzyresortowy skład PKAPI stwarza przesłanki do utworzenia wspólnej dyspozycji mocą komputerów i społecznej kontroli realizacji tego przedsięwzięcia. Istota idei wspólnego dysponowania mocą powinna zapewnić pełne wykorzystanie wszystkich maszyn w ciągu dwuzmianowej pracy, a jednocześnie w wypadku awarii pojedynczych maszyn — zapewniać uruchamianie trzeciej zmiany. Jeżeli np. komputery w ZETO Łódź w ciągu określonego dnia są przeciążone pracą, która nie mieści się w czasie obejmującym trzy zmiany, a w EKORNO lub MPK maszyny te w danym dniu mogą być maksymalnie obciążone przez jedną lub półtorej zmiany, wówczas zostałyby one dociążone częścią prac spiętrzonych w ZETO.

Poza tym wspólne dysponowanie mocą przewidywałoby również bieżącą pomoc w zakresie usuwania awarii, udostępnianie wzajemne posiadanych części maszyn, co jest szczególnie uzasadnione słabym zapleczem ELWRO co do wyposażenia w części zamienne oraz dysponowania odpowiednio licznego i właściwie przygotowanego personelu do świadczenia usług serwisowych.

Wprowadzenie wspomnianej koncepcji do praktyki powinno mieć szanse powodzenia choćby ze względu na olbrzymie koszty ponoszone przez państwo z tytułu instalowania komputerów. Jak dotąd, można przyjąć w przybliżeniu, że np. na terenie Łodzi zainwestowano kwotę około 100 mln zł w sam tylko zakup komputerów.

Okoliczności takie zobowiązują do wytypowania przez zainteresowane placówki takich ludzi, których cechowałoby głębokie poczucie odpowiedzialności, wzajemne zrozumienie i życzliwość. Jest to zasadniczy warunek powodzenia tej słusznej koncepcji. Bowiem niezależnie od tego, z jak doskonałą techniką — tutaj mówimy o elektronicznej technice obliczeniowej — mamy do czynienia, ogniwem decydującym jest człowiek.

Doniosłość rozwoju informatyki w kraju, wynikająca z potrzeb lepszego gospodarowania akcentowana jest ze szczególną mocą przez najwyższe władze państwowe, co stwarza właściwą płaszczyznę do wykorzystania każdej twórczej inicjatywy społecznej i zobowiązuje wszystkich działaczy gospodarczych jako osoby bezpośrednio odpowiedzialne do tego, aby problem tu przedstawiony mógł zostać rzeczowo i z należytą troską rozwiązany.

Należy z satysfakcją podkreślić, że zbliżone do sformułowanych tu sugestie zawarte są w aktualnych zaleceniach i koncepcjach Krajowego Biura Informatyki, które szczególnie silnie podkreśla potrzebę tworzenia przez ZETO regionalnych ośrodków serwisowych.

Pro domo sua, ZETO Łódź zobowiązuje się podjąć jak najszybciej tego typu inicjatywy.

Zygmunt Łuczak
ZETO Łódź

Inż. Zygmunt Topolewski
Wrocław
ul. Skwierzyńska 38 m. 7

Redakcja „Informatyki"
Warszawa

Od kilku miesięcy w niektórych ośrodkach obliczeniowych zostały zainstalowane i uruchomione maszyny cyfrowe ODRA 1304. Równolegle z instalacją i uruchomianiem maszyn rozpoczęto szkolenie konserwatorów, programistów i operatorów ODRY 1304. Wydane zostały również komplety podręczników dotyczących budowy jednostki centralnej, urządzeń wejścia—wyjścia, testowania maszyny oraz podręczniki programistów i operatorów. Jednym z mankamentów jest fakt, że do obecnej pory nie ukazała się książka „Programy biblioteczne", oznaczona w spisie numerem 13007, której dotkliwy brak odczuwa się we wszystkich ośrodkach posiadających mc ODRA 1304, drugim — brak ujednoliconego słownictwa (nazewnictwa) polskiego związanego z nową maszyną, z którą należy komunikować się w języku angielskim. Z tej przyczyny powstała więc cała masa niejasności. Podręcznik operatora częściowo tę sprawę rozwiązuje w dziale „komunikaty wprowadzone", natomiast mniej w dziale „komunikaty wyprowadzone", gdyż np. komunikat: S # prog; UNIT q : — USED AS X posiada objaśnienie: „Komunikat ten jest wynikiem wykonania rozkazu 156 (ALLOT), który żądał przydzielenia odpowiedniego typu urządzenia zewnętrznego jako jednostki programowej X". Chyba trochę za mało! Przecież kto nie zna języka angielskiego nadal nie wie co to jest „USED AS X", no i w końcu czy maszyna przydzieliła sobie to urządzenie, czy też nie? W każdym razie są pewne wątpliwości. Takich i podobnych komunikatów jest cała masa lub nadania dwom różnym słowom tego samego znaczenia, jak np. WRONG i DELETE, tłumaczy podręcznik operatora jako SKREŚL.

Ogólnie rzecz biorąc brak jest słownika tłumaczącego wszystkie komunikaty wprowadzone, jak i wyprowadzone, który by obowiązywał we wszystkich ośrodkach obliczeniowych i likwidował wszelakie wątpliwości przy kontakcie z maszyną.

Człowiek ma to do siebie, że tam, gdzie występują jakieś braki czy niedomówienia zaczyna sobie po swojemu określać pewne zagadnienia czy słowa konieczne do wykonania jakiejś czynności. W ten sposób zarówno na kursach, jak i w ośrodkach obliczeniowych stworzono żargon, który zaczyna nabierać prawa obywatelstwa. Przytoczę kilka przykładów z tegoż żargonu. Ponieważ pragnę przekazać te określenia w brzmieniu dosłownym, będę je pisał jak są wymawiane:

- proszę ten program WYDELEJTOWAĆ
- proszę te taśmy ZASKRECZOWAĆ
- czy już nastąpiło ZAKLOZETOWANIE?
- proszę założyć SKRECZ — TAŚMĘ
- wymagane HARDWARE
- proszę ten program ZDAMPOWAĆ
- proszę to urządzenie WRONGOWAĆ itp.

Nie będę przytaczał mniej rażących określeń, jak BUSY, FAIL, PINTY, SOWTWARE i inne.

Podana przez mnie angielszczyzna zaczyna, jak już wspomniałem, nabierać prawa obywatelstwa. Ponadto ludzie wolą używać np. bliżej nie określonego słowa HARDWARE i umieszczać w tym punkcie wszystko, co dotyczy zarówno jednostki centralnej, jak i urządzeń zewnętrznych niż zastanowić się nad rozdziałem tych zagadnień.

Również brak jest jednolitego określenia nazwy SCRATCH TAPE. Jedni określają ją jako taśmę roboczą, inni jako taśmę zaetykietowaną. Ponieważ są rozbieżności zdań, użyto więc w języku potocznym TAŚMA ZASKRECZOWANA i wówczas każdy niech sobie tłumaczy na swój sposób.

Uważam, że tych kilka przykładów dostatecznie obrazuje, jak bardzo została okaleczona mowa polska w związku z wprowadzeniem mc ODRA 1304. Nie będę już rozwodził się nad tym, że są i tacy, którzy lubują się w używaniu obcych (spolszczonych) określeń.

Przypuszczam, że nie od rzeczy byłoby, gdyby razem z wprowadzeniem nowych (obcych) urządzeń ukazywał się obowiązujący słownik oznaczeń polskich stosowanych przy obsłudze danej maszyny czy urządzenia. Nie możemy czekać na dokonanie takiego tłumaczenia przez Anglików. Miałem możność oglądać instrukcję obsługi obrabiarki w Fabryce Urządzeń Mechanicznych we Wrocławiu napisanej przez Anglików. Oto fragmenty tej instrukcji:

NAPSZUT, DO TYPU, PRENKO, ZACZONCI, NAZENIDZIE, PEDAŁO ZARINACZONY...

Szanując dobrą wolę ludzi, którzy opracowali wspomnianą instrukcję należy się jednak wystrzegać podobnego tłumaczenia.

Nie może być więc rzeczą obojętną troska o czystość języka polskiego. Mamy przykre doświadczenia z okresu wprowadzania pierwszych urządzeń mechanicznych, których nazwy, takie jak „hebel, żaga, majsel, waserwaga" i inne pokutują w naszym słownictwie do obecnej pory. Niechaj więc doba automatyzacji nie doprowadzi do dalszego zachwaszczania języka polskiego.

Z KRAJOWEGO BIURA INFORMATYKI

PIERWSZY KOMPUTER W PRZEMYŚLE LEKKIM

Z okazji Święta 22 Lipca 1971 r. nastąpiło przekazanie do eksploatacji komputera ODRA 1304 w Branżowym Ośrodku Obliczeniowym utworzonym przy Centralnym Laboratorium Przemysłu Wełnianego Południe w Bielsku-Białej. Komputer mieści się w nowocześnie zaprojektowanych i wykończonych, klimatyzowanych pomieszczeniach o powierzchni 1200 m², które można zaliczyć do najbardziej funkcjonalnych w kraju. Od chwili podjęcia decyzji w sprawie zakupu komputera do chwili przekazania go do eksploatacji upłynęło 1,5 roku, co w krajowych warunkach jest okresem rekordowym.

Aktualnie w Ośrodku pracuje 48 wysoko kwalifikowanych pracowników, realizujących pełne cykle przygotowania i wdrażania systemów przetwarzania danych dla przemysłu wełnianego.

Doświadczenia z lat ubiegłych uzyskane przez Ośrodek na komputerach IBM 1400 w Warszawie, ICL 4-50 i MIŃSK 22 w Katowicach pozwoliły do chwili obecnej przygotować do realizacji i wdrożyć na komputerze ODRA 1304 trzy systemy:

- ewidencji, rozliczenia i analizy tkanin gotowych
- ewidencji i analizy zużycia przędzy
- rozliczania transakcji giełdowych.

W najbliższym okresie — do końca roku 1972 — na komputerze będą realizowane wszystkie rozliczenia sfery zaopatrzenia produkcji i zbytu przedsiębiorstw zgrupowanych w Zjednoczeniu Przemysłu Wełnianego-Południe.

Przy okazji otwarcia Ośrodka odbyło się dnia 20 lipca br. ogólnokrajowe seminarium informatyków resortu przemysłu lekkiego. M.in. przedstawiono założenia systemu bieżącego kierowania produkcją: USTER-MINITEX w Zakładzie KENTEX w Kętach. Do każdego krosna zostanie podłączony czujnik, który będzie przesyłać do mini-komputera bieżące informacje o przebiegu produkcji. Do roku 1975 zamierza się zaadaptować ten system w kilku zakładach przemysłu wełnianego tak, aby do roku 1980 wszystkie zakłady Zjednoczenia były kierowane w hierarchicznym systemie informatycznym, wyposażonym w sieć mini-komputerów produkcji krajowej.

WIADOMOŚCI ZE ŚLĄSKA

- Przeprowadzono w dniu 21 lipca 1971 roku w Katowicach uzgodnienia z Dyrekcją OPT w sprawie uruchomienia Ośrodka Doskonalenia Kadr Informatyki. Uzgodniono wstępnie:

a) szkolenie 1200 osób w ciągu roku w czterech turnusach równoległych po 30 osób (wymaga to dziennie 8 wykładowców)

b) zapewnienie bazy noclegowej w miejscowym hotelu i hotelu PTTK (docelowo własny hotel na 400 miejsc)

c) oddanie dla informatyki jednego pawilonu oraz pomieszczeń na laboratoria (5000 m²) w 2 pawilonach wystawowych, gdzie są obecnie zainstalowane komputery MIŃSK 22 i MIŃSK 32 (w skrzyniach); pomieszczenie jest klimatyzowane i przystosowane do ETO

d) przydzielenie OPT 24 etatów poprzez Zjednoczenie Informatyki, w tym: 10 dla stałych wykładowców (rotacja co 2 lata); 10 dla obsługi sprzętu informatycznego; 4 dla organizacji i programowania kursów

e) utworzenie Działu Szkolenia Informatyków (24 osoby j.w.) w Zakładzie Aktualizacji Wiedzy

f) koszt przeszkolenia 1 słuchacza szacowany na 10—15 tys. zł pokrywany będzie z nakładów inwestycyjnych użytkowników (KBI ma pisemne uzgodnienie z Ministerstwem Finansów); da to obrót OPT około 20 mln zł rocznie

g) możliwość przydzielenia w latach 1976—1980 około 7 mln zł na rozbudowę ośrodka informatyki

h) możliwość przydzielenia w latach 1971—1975 do 2 mln zł na rozbudowę zaplecza socjalno-bytowego dla słuchaczy kursów informatyki szkolenia stacjonarnego, bez którego trudno byłoby prowadzić szkolenie

i) możliwość przekazania środków w granicach 6 mln zł na rozbudowę bazy poligraficznej wydawnictw informatyki oraz sprzętu dydaktycznego zorientowanego na informatykę.

Wymienione środki mieszczą się w limitach inwestycyjnych Zjednoczenia Informatyki

j) podpisanie porozumienia w wyżej omówionych sprawach pomiędzy OPT i KBI nastąpi w roku bieżącym.

- Przeprowadzono konsultację w Centralnym Biurze Rozliczeń Przemysłu Węglowego w Katowicach w zakresie realizacji zadania węzłowego. Uzgodniono współpracę systemu „węgla" z systemem „transportu" w zakresie harmonogramowania marszrut transportowych. Przedyskutowano i uzgodniono postępowanie w sprawie zakupu sprzętu.

SPECJALIŚCI Z ZOWAR ZWIEDZILI CZECHOSŁOWACKIE OŚRODKI OBLICZENIOWE

Grupa specjalistów ZOWAR-u pod kierownictwem zastępcy dyrektora do spraw przetwarzania, mgr inż. K. Szula, zwiedziła w dniach 30 maja ÷ 4 czerwca cztery czechosłowackie ośrodki obliczeniowe. Oto garść spostrzeżeń zaczerpniętych ze sprawozdania.

Ośrodek Obliczeniowy Przemysłu Chemicznego VUTE — CHP — Praha, wyposażony jest w komputery IBM 360/40 i IBM 360/20. Ośrodek realizuje prace obliczeniowe z zakresu przetwarzania danych dla przedsiębiorstw zlokalizowanych w rejonie Pragi oraz dla dwóch użytkowników spoza tego rejonu, z wykorzystaniem teletransmisji. Głównym kierunkiem zastosowań jest obrót towarowy realizowany w oparciu o własny system rozliczeń. Drugą dziedziną zastosowań jest planowanie wyrobów w oparciu o system IBM-PICS (Production Information and Control System).

Ośrodek Obliczeniowy Zakładu Produkcji Samochodów SKODA ma charakter ośrodka zakładowego. Zlokalizowany jest na terenie Zakładu i pracuje dla jego potrzeb. Zakład posiada długoletnie tradycje stosowania maszyn analitycznych, które wykorzystywane są nadal. Szczupły zestaw maszyny IBM 360/30 i b. mała pamięć wewnętrzna ograniczają możliwość wykorzystania programów producenta (IBM). Jak dotąd, udało się wprowadzić do eksploatacji użytkowej część systemu sterowania produkcją — PICS. Zadaniem perspektywicznym ośrodka jest rozszerzenie dotychczas eksploatowanego systemu o nowe moduły PICS-u, a szczególnie o planowanie produkcji netto.

Ośrodek Obliczeniowy Komisji Planowania CSRS ma charakter typowo usługowy. Obsługuje około 500 użytkowników. Zwraca uwagę bardzo dobra organizacja pracy oparta na małej liczbie pracowników własnych i rozwijaniu kadry specjalistów u użytkowników. Ośrodek wyposażony jest w trzy komputery: UNIVAC 1040, IBM 1040 i IBM 360/40. Prace ośrodka obejmują bardzo zróżnicowaną tematykę, przy czym większość prac nosi charakter jednorazowy. Rozpoczęto prace projektowe zmierzające do zbudowania ogólnokrajowego systemu informatycznego z centralnym bankiem danych, dla potrzeb administracji państwowej.

Ośrodek Obliczeniowy przy Biurze Projektowym Prac Inżynieryjno-Budowlanych IPS w Pradze realizuje prace obliczeniowe z zakresu ewidencji i rozliczeń włącznie z fakturami oraz kosztorysowania dla planowanych obiektów (80% czasu), a także przeprowadza obliczenia inżynieryjno-techniczne dla potrzeb projektantów. Ośrodek wyposażony jest w komputer IBM 360/30 z pamięcią operacyjną 32 K/Bajtów. Rozpoczęto prace nad stworzeniem banku danych, który w przyszłości będzie bazą dla systemu informatycznego obejmującego całość zagadnień budownictwa w CSRS.

Omawiane sprawozdanie uzupełniają spostrzeżenia i wnioski wynikające z porównania zwiedzonych ośrodków czechosłowackich do ośrodków polskich.

- Poziom wykonywanych prac projektowych nie jest wyższy od analogicznych projektów wykonanych w ZOWAR. Jednakże zwiedzone ośrodki dysponują bogatszym zapleczem technicznym zarówno w zakresie samych komputerów jak i sprzętu i urządzeń pomocniczych.

Cz.

● Warunki pracy w ośrodkach CSRS są znacznie lepsze. Dysponują one z reguły dużą powierzchnią użytkową, pomieszczenia są zaplanowane funkcjonalnie, zwraca uwagę stosunkowo bogate wyposażenie biur w urządzenia pomocnicze.

● Organizacja pracy ośrodków CSRS stoi na wyższym poziomie. Współpraca ośrodków z klientami w większości przypadków ogranicza się do udzielania konsultacji, w związku z czym liczebność kadry projektantów i programistów jest znacznie mniejsza niż w podobnych ośrodkach w Polsce. Klient sam organizuje i opracowuje swój system APD, a rola ośrodka obliczeniowego sprowadza się do jego eksploatacji.

● Celowe wydaje się rozważenie przebudowy organizacji pracy w polskich ośrodkach obliczeniowych, przy zwróceniu większej uwagi na odpowiednie przygotowanie klientów. (cz)

POWOŁANIE

PAŃSTWOWEJ RADY INFORMATYKI

Prezes Rady Ministrów powołał zarządzeniem z dnia 1 września 1971 roku PAŃSTWOWĄ RADĘ INFORMATYKI jako organ opiniodawczo-doradczy Rady Ministrów.

Do zakresu działania Rady należy:

● dokonywanie oceny informatyki w kraju oraz opiniowanie kierunków dalszego jej rozwoju

● opiniowanie programów oraz planów wieloletnich i rocznych rozwoju informatyki w gospodarce narodowej

● opiniowanie i wnioskowanie w sprawach rozwoju krajowej produkcji sprzętu informatyki oraz zamierzeń w zakresie jego importu i eksportu

● opiniowanie i wnioskowanie kierunków prac naukowo-badawczych, projektowych i konstrukcyjnych w dziedzinie informatyki

● opiniowanie i wnioskowanie w sprawach programów i kierunków szkolenia i doskonalenia fachowych kadr dla informatyki

● opiniowanie projektów aktów prawnych o podstawowym znaczeniu dla rozwoju informatyki

● opiniowanie spraw z zakresu informatyki przekazanych przez zainteresowanych ministrów (kierowników urzędów centralnych).

W skład RADY wchodzą: przewodniczący, którym jest z urzędu przewodniczący Komitetu Nauki i Techniki, sekretarz oraz 30 członków powoływanych przez prezesa Rady Ministrów na wniosek przewodniczącego Komitetu Nauki i Techniki.

PAP, ‚Życie Warszawy' z dnia 11.IX.1971 r.